

**HALAMAN JUDUL**  
**APLIKASI STEGANOGRAFI PENYEMBUNYI PESAN**  
**SUARA TERENKRIPSI BERBASIS ANDROID**



**SKRIPSI**

**Diajukan untuk memenuhi salah satu syarat guna mencapai gelar**

**Sarjana Komputer pada Jurusan Teknik Informatika**

**Fakultas Sains dan Teknologi**

**UIN Alauddin Makassar**

**Oleh:**

**MUAWAD ULUL AZMI**

**NIM: 60200110049**

**FAKULTAS SAINS DAN TEKNOLOGI**

**UIN ALAUDDIN MAKASSAR**

**2015**

## **PERSETUJUAN PEMBIMBING**

Pembimbing penulisan skripsi saudara **Muawad Ulul Azmi: 60200110049**, mahasiswa Jurusan Teknik Informatika pada Fakultas Sains dan Teknologi Universitas Islam Negeri (UIN) Alauddin Makassar, setelah dengan seksama meneliti dan mengoreksi skripsi yang bersangkutan dengan judul, **“Aplikasi Steganografi Penyembunyi Pesan Suara Terenkripsi Berbasis Android”**, memandang bahwa skripsi tersebut telah memenuhi syarat-syarat ilmiah dan dapat disetujui untuk diajukan ke sidang Munaqasyah.

Demikian persetujuan ini diberikan untuk proses selanjutnya.

Makassar, September 2015



**Pembimbing I**

**Ridwan A.Kambau, S.T., M.Kom.**

**NIP. 19740105 200801 1 006**

**Pembimbing II**

**Faisal, S.T., M.T.**

**NIP. 19720721 201101 1 001**

## **PERNYATAAN KEASLIAN SKRIPSI**

Saya yang bertanda tangan di bawah ini :

Nama : Muawad Ulul Azmi

NIM : 60200110049

Tempat/Tgl. Lahir : Jakarta, 12 Juli 1992

Jurusan : Teknik Informatika

Fakultas/Program : Sains dan Teknologi

Judul : Aplikasi Steganografi Penyembunyi Pesan Suara  
Terenkripsi Berbasis Android

Menyatakan dengan sebenarnya bahwa skripsi yang saya tulis ini benar merupakan hasil karya saya sendiri. Jika kemudian hari terbukti bahwa ini merupakan duplikasi, tiruan, plagiat, atau dibuat oleh orang lain, sebagian atau seluruhnya, maka skripsi dan gelar yang diperoleh karenanya batal demi hukum.

Makassar, September 2015

Penyusun,

**Muawad Ulul Azmi**

**NIM : 60200110061**

## PENGESAHAN SKRIPSI

Skripsi yang berjudul “*Aplikasi Steganografi Penyembunyi Pesan Suara Terenkripsi Berbasis Android*” yang disusun oleh Muawad Ulul Azmi, NIM 60200110049, mahasiswa Jurusan Teknik Informatika pada Fakultas Sains dan Teknologi UIN Alauddin Makassar, telah diuji dan dipertahankan dalam sidang *munaqasyah* yang diselenggarakan pada Hari Rabu, Tanggal 16 September 2015 M, bertepatan dengan 3 Dzhul-Hijjah 1436 H, dinyatakan telah dapat diterima sebagai salah satu syarat untuk memperoleh gelar Sarjana dalam Ilmu Teknik Informatika, Jurusan Teknik Informatika.

Makassar, 16 September 2015 M.

3 Dzhul-Hijjah 1436 H.

### DEWAN PENGUJI :

Ketua	:	Prof. Dr. H. Arifuddin Ahmad, M. Ag.	(.....)
Sekretaris	:	Yusran Bobihu, S.Kom., M.Si.	(.....)
Munaqisy I	:	Dr. H. Kamaruddin Tone, M.M.	(.....)
Munaqisy II	:	Mega Orina Fitri, S.T., M.T.	(.....)
Munaqisy III	:	Dr. Sohra, M.Ag.	(.....)
Pembimbing I	:	Ridwan Andi Kambau, S.Kom., M.T.	(.....)
Pembimbing II	:	Faisal, ST., M.T.	(.....)

Diketahui oleh :

Dekan Fakultas Sains dan Teknologi

UIN Alauddin Makassar,

**Prof. Dr. H. Arifuddin Ahmad, M. Ag.**

NIP. 19691205 199303 1 001

## KATA PENGANTAR



Puji syukur Alhamdulillah penulis panjatkan kepada Allah swt. dengan segala rahmat, hidayah, dan ilmu-Nya sehingga penulis dapat menyelesaikan skripsi ini, yang merupakan salah satu persyaratan untuk mendapatkan gelar sarjana di Fakultas Sains dan Teknologi Jurusan Teknik Informatika. Tidak lupa penulis kirimkan shalawat dan salam kepada Nabi Muhammad saw yang telah menuntun kita menjadi manusia yang lebih berperadaban dan saling menghargai.

Banyak kendala yang dihadapi dalam proses penyusunan skripsi ini, dengan segala keterbatasan yang dimiliki penulis maka skripsi ini tidak akan mungkin selesai tanpa bantuan berbagai pihak, baik pihak keluarga, pihak fakultas, maupun teman-teman, juga berkat usaha, do'a, dan waktu yang telah diluangkan oleh penulis, hingga akhirnya skripsi ini dapat diselesaikan. Olehnya itu, pada kesempatan kali ini penulis mengucapkan terima kasih yang tak terhingga kepada seluruh pihak yang telah membantu untuk menyelesaikan skripsi ini, terutama kepada ayahanda, Prof. Dr. Sabri Samin, M. Ag. yang selalu memotivasi dan memberikan ilmu yang bermanfaat kepada penulis dan Alm. Ibunda Zawil Marhamah, S.Pd. (alm) yang tak pernah lelah dan ikhlas membesarkan penulis, mulai dari melahirkan, mendidik bahkan disaat almarhumah sedang sakit.

1. Rektor Universitas Islam Negeri (UIN) Alauddin Makassar, Prof. Dr. H. Musafir Pababbari, M. Si..
2. Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri (UIN) Alauddin Makassar, Prof. Dr. H. Arifuddin Ahmad, M. Ag..
3. Ketua Jurusan Teknik Informatika, Faisal, S.T, M.T. dan Sekretaris Jurusan Teknik Informatika, Mega Orina Fitri, S.T., M.T..
4. Pembimbing I, Ridwan Andi Kambau, S.T., M.Kom. dan pembimbing II, Faisal, S.T, M.T. yang telah membimbing dan membantu untuk mengembangkan pemikiran dalam penyusunan skripsi hingga selesai.
5. Seluruh dosen, staf dan karyawan Jurusan Teknik Informatika Fakultas Sains dan Teknologi UIN Alauddin Makassar yang telah memberikan sumbangsih baik tenaga maupun pikiran.
6. Saudaraku Bunyanun Marsus, S.T. dan saudariku Ns. Uzlifah Zulmi, S.Kep. yang senantiasa memberikan dukungan dan dorongan untuk menyelesaikan skripsi ini.
7. Muhammad Hade Mopilie, S.Kom, Ahmad Kautsar, S.Kom yang telah banyak memberikan ide, referensi dan bantuan.
8. Kakanda Iin Dwi Ayuningsih, S.Kom. yang telah memberikan referensi yang sangat bermanfaat.

9. Teman seperjuanganku Mudassir, Muhammad Irsan Karapa, S.Kom dan sahabat – sahabat B10S dari Teknik Informatika Angkatan 2010 yang telah menjadi saudara/i seperjuangan dalam suka dan duka bersama dalam menempuh pendidikan di kampus.
10. Kepada kakak-kakak, adik-adik dan teman-teman di *study club* Exomatik yang telah memberikan banyak ilmunya selama kuliah di kampus
11. Meinarti Arini Pusparani, S.Kom. yang telah setia menemani suka dan duka dalam menyelesaikan skripsi ini.
12. Kepada Clan COC Mahasiswa Tua, yang telah memberikan banyak hiburan di saat penulis merasa pusing dengan skripsi yang dikerjakan.
13. Semua pihak yang tidak dapat disebutkan satu per satu, yang telah dengan tulus ikhlas memberikan doa dan motivasi sehingga skripsi ini dapat terselesaikan.

Penulis menyadari, masih banyak kekurangan dalam skripsi ini karena keterbatasan yang dimiliki penulis sebagai manusia yang tak luput dari kesalahan. Kritik dan saran yang membangun dari berbagai pihak demi perbaikan dan penyempurnaan akan penulis terima dengan senang hati. Semoga skripsi ini dapat berguna bagi para pembaca atau siapa saja yang tertarik dengan materinya. Lebih dan kurangnya penulis mohon maaf yang sebesar-besarnya, semoga Allah swt melimpahkan rahmat-Nya kepada kita semua. Aamiin.

Makassar, September 2015

Muawad Ulul Azmi

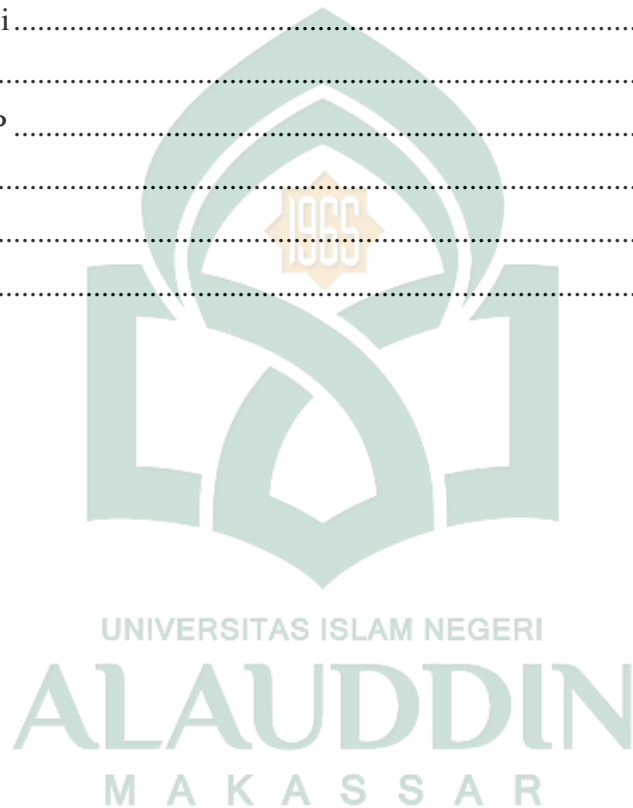




## DAFTAR ISI

PERSETUJUAN PEMBIMBING.....	i
PERNYATAAN KEASLIAN SKRIPSI.....	ii
KATA PENGANTAR .....	iv
DAFTAR ISI.....	viii
DAFTAR GAMBAR .....	x
DAFTAR TABEL.....	xii
ABSTRAK .....	xiii
BAB I PENDAHULUAN .....	1
A. Latar Belakang Masalah.....	1
B. Rumusan Masalah .....	4
C. Fokus Penelitian dan Deskripsi Fokus .....	4
D. Kajian Pustaka.....	5
E. Tujuan dan Kegunaan Penelitian .....	8
BAB II TINJAUAN TEORETIS .....	10
A. Aplikasi .....	10
B. Keamanan.....	12
C. Android .....	22
BAB III METODOLOGI PENELITIAN.....	34
A. Jenis dan Lokasi Penelitian.....	34
B. Pendekatan Penelitian .....	34
C. Sumber Data.....	35
D. Metode Pengumpulan Data.....	35
E. Instrumen Penelitian.....	36
F. Teknik Pengolahan dan Analisis Data .....	37
G. Metode Perancangan Aplikasi.....	38
H. Teknik Pengujian Sistem.....	38
I. Rancangan Tabel Pengujian.....	39

BAB IV ANALISIS DAN PERANCANGAN SISTEM .....	40
A. Analisis Sistem yang Sedang Berjalan.....	40
B. Analisis Sistem yang diusulkan .....	41
C. Perncangan Sistem .....	43
BAB V IMPLEMENTASI DAN ANALISIS HASIL .....	54
A. Implementasi .....	54
B. Pengujian.....	57
BAB VI PENUTUP .....	70
A. Kesimpulan .....	70
B. Saran.....	70
Daftar Pustaka .....	72



## DAFTAR GAMBAR

Gambar II.1 : Proses Kriptografi (Arif, 2014) .....	14
Gambar II.2 : Gambar yang belum dienkrpsi (Merdeka, 2013).....	14
Gambar II.3 : Gambar setelah dienkrpsi (Merdeka, 2013) .....	15
Gambar II.4 : Contoh Steganografi (Merdeka, 2013).....	22
Gambar II.5 : Arsitektus OS Android (Donny, 2011).....	24
Gambar II.6 : Siklus Activity Android.....	27
Gambar IV.1 : <i>Flowmap</i> sistem yang berjalan.....	40
Gambar IV.2 : <i>use case diagram</i> .....	43
Gambar IV.3 : <i>Class Diagram</i> .....	44
Gambar IV.4 : <i>sequence diagram menu encoding</i> .....	45
Gambar IV.5 : <i>sequence diagram menu decoding</i> .....	45
Gambar IV.6 : <i>Activity Diagram</i> .....	46
Gambar IV.7 : Struktur Navigasi .....	47
Gambar IV.8 : Desain antarmuka <i>splash screen</i> .....	48
Gambar IV.9 : Desain antarmuka menu utama.....	48
Gambar IV.10 : Desain antarmuka menu <i>encoding</i> .....	49
Gambar IV.11 : Desain antarmuka menu decoding.....	51
Gambar IV.12: Desain antarmuka menu panduan .....	52
Gambar IV.13 : Desain antarmuka menu profil.....	53
Gambar V.1 : Antarmuka <i>splashscreen</i> . ....	54
Gambar V.2 : Antarmuka menu <i>encoding</i> .....	55
Gambar V.3 : Antarmuka menu <i>decoding</i> . ....	56
Gambar V.4 : Antarmuka menu panduan. ....	56
Gambar V.5 : Antarmuka menu profil. ....	57
Gambar V.6 : Hasil pengujian <i>button load</i> suara.....	60
Gambar V.7 : Hasil pengujian <i>button</i> enkripsi.....	60
Gambar V.8 : Hasil pengujian <i>image view</i> .....	61

Gambar V.9 : Hasil pengujian <i>button</i> sisipkan .....	62
Gambar V.10 : Hasil pengujian <i>button</i> simpan.....	62
Gambar V.11 : Hasil pengujian <i>button</i> bagikan <i>via bluetooth</i> .....	63
Gambar V.12 : Hasil pengujian <i>button</i> bagikan <i>via BBM</i> .....	63
Gambar V.13 : Hasil pengujian <i>image view</i> .....	65
Gambar V.14 : Hasil pengujian <i>button</i> ekstrak.....	65
Gambar V.15 : Hasil pengujian <i>button</i> dekripsi.....	66
Gambar V.16 : Hasil pengujian <i>button</i> simpan.....	66
Gambar V.17 : Hasil pengujian <i>tab</i> panduan.....	67
Gambar V.18 : Hasil pengujian <i>tab</i> profil.....	68



## DAFTAR TABEL

Tabel II.1 Daftar Simbol <i>Use Case Diagram</i> .....	30
Tabel II.2 Daftar Simbol <i>Class Diagram</i> (Rosenberg, 2007) .....	31
Tabel II.3 Daftar Simbol <i>Sequence Diagram</i> (Rosenberg, 2007) .....	32
Tabel III.1 : kasus yang akan diuji .....	39
Tabel V.1 : Tabel pengujian menu <i>encoding</i> .....	58
Tabel V.2 : Tabel menu <i>decoding</i> .....	64
Tabel V.3 : Tabel menu panduan .....	67
Tabel V.4 : Tabel menu panduan .....	68
Tabel V.4 : Tabel hasil encoding .....	69



## ABSTRAK

**Nama** : Muawad Ulul Azmi  
**Nim** : 60200110049  
**Jurusan** : Teknik Informatika  
**Judul** : Aplikasi Steganografi Penyembunyi Pesan Suara Terenkripsi Berbasis Android  
**Pembimbing I** : Drs.H. Kamaruddin Tone, M.M.  
**Pembimbing II** : Faisal, S.T., M.T.

---

Penelitian ini dilatarbelakangi karena seringnya terjadi kasus penyadapan atau pembajakan data dalam proses pengirimannya. Karena itu dibutuhkan sebuah keamanan untuk mengamankan data tersebut. Ada dua teknik yang digunakan dalam mengamankan data, yaitu kriptografi dan steganografi. Kriptografi adalah teknik mengamankan pesan dengan cara mengacak atau merubah isi pesan dengan algoritma tertentu sedangkan steganografi adalah teknik mengamankan pesan dengan cara menyembunyikan pesan ke dalam media lain. Dengan kedua teknik tersebut akan dibuat aplikasi yang akan mengamankan data, dalam hal ini adalah data yang berupa pesan suara.

Aplikasi steganografi penyembunyi pesan suara terenkripsi merupakan aplikasi yang digunakan untuk mengamankan pesan suara singkat, seperti hasil wawancara. Aplikasi ini menggunakan dua teknik untuk mengamankan datanya, untuk kriptografinya menggunakan algoritma *serpent* dan metode *end of file* untuk proses steganografinya.

Dalam melakukan penelitian ini, jenis penelitian yang digunakan adalah penelitian lapangan kualitatif eksperimental. Aplikasi ini berjalan pada perangkat *mobile* dengan sistem operasi Android dan dirancang menggunakan bahasa pemrograman Java. Pembangunan aplikasi ini menggunakan *software* Eclipse. Dalam pemodelannya aplikasi ini menggunakan metode *UML* berupa *use case diagram*, *class diagram* dan *sequence diagram* dan diuji menggunakan metode pengujian *black box*. Hasil dari penelitian ini adalah aplikasi steganografi penyembunyi pesan suara terenkripsi, Aplikasi ini dibuat sebagai sarana untuk pengamanan pesan suara.

**Kata Kunci:** *Android, Steganografi, Kriptografi, Java, UML*

# BAB I

## PENDAHULUAN

### *A. Latar Belakang Masalah*

Keamanan merupakan hal terpenting dalam kehidupan manusia begitu pula dalam perkembangan teknologi. Keamanan adalah keadaan bebas dari bahaya, dimana istilah ini biasa digunakan dengan hubungan kepada kejahatan, kecelakaan dan lain-lain. Rasa aman tidak bisa digambarkan jika tidak dalam situasi ketatukan. Dalam konteks teknologi informasi dan komunikasi, keamanan merupakan salah satu aspek penting yang perlu diperhatikan. Terjadinya kasus pembajakan, penyadapan ataupun pencurian data adalah hal yang harus ditakutkan, karena ada data yang sifatnya rahasia. Oleh karena itu, data tersebut perlu diamankan dari gangguan orang-orang yang tidak bertanggung jawab. Allah swt mengkaruniakan manusia akal dan pikiran, oleh karena itu manusia wajib untuk mempelajari masalah keamanan. Sebagaimana firman Allah swt dalam Q.S Al-Hasyr/59 : 23 :

هُوَ اللَّهُ الَّذِي لَا إِلَهَ إِلَّا هُوَ الْمَلِكُ الْقُدُّوسُ السَّلَامُ الْمُؤْمِنُ الْمُهَيْمِنُ الْعَزِيزُ الْجَبَّارُ

الْمُتَكَبِّرُ سُبْحَانَ اللَّهِ عَمَّا يُشْرِكُونَ ﴿٢٣﴾

Terjemahnya :

*Dialah Allah yang tiada Tuhan selain Dia, Raja, yang Maha Suci, yang Maha Sejahtera, yang Mengaruniakan Keamanan, yang Maha Memelihara, yang Maha perkasa, yang Maha Kuasa, yang memiliki segala Keagungan, Maha suci Allah dari apa yang mereka persekutukan.*(Departemen Agama, 2008)

Kata al-mu'min (yang mengaruniakan keamanan) terambil dari kata amina, yang melahirkan sekian banyak bentuk antara lain iman, amanah dan aman. Amanah adalah lawan dari khianat yang melahirkan ketenangan batin serta rasa aman karena adanya membenaran dan kepercayaan terhadap sesuatu, sedang iman adalah membenaran hati dan kepercayaan terhadap sesuatu. Sedangkan *al-muhaimin* (yang maha memelihara) adalah sebagai yang menjadi saksi terhadap sesuatu serta memeliharanya. Dari penjelasan dua kata tersebut, maka implementasi dari karunia Allah swt dari segi keamanan, maha mengaruniakan keamanan dan maha memelihara, dapat dilakukan dengan membuat sebuah aplikasi yang dapat mengamankan data.

Seiring perkembangan teknologi yang begitu pesat, ancaman terhadap keamanan data dan informasi yang dibutuhkan semakin besar, terutama terhadap data atau informasi yang bersifat rahasia. Berbagai ancaman di dunia maya seperti *hacker* atau *cracker* mengkhawatirkan masyarakat dalam proses pengiriman data, apalagi jika data tersebut bersifat rahasia sehingga dapat menghambat proses pengiriman data tersebut karena adanya ancaman tersebut. Dengan alasan itu, maka dibutuhkan suatu



cara untuk mengamankan data tersebut. Contoh metode yang dapat digunakan untuk melindungi data yaitu kriptografi dan steganografi.

Kriptografi merupakan teknik mengamankan data dengan cara mengacak atau merubah isi datanya dengan menggunakan algoritma tertentu, sedangkan steganografi adalah teknik mengamankan data dengan cara menyembunyikan data kedalam data yang lain. Perbedaan mendasar pada kedua teknik di atas adalah saat data tersebut telah diamankan dengan salah satu teknik di atas. Pada kriptografi, setelah pesan diamankan masih akan menimbulkan kecurigaan terhadap data karena masih dapat terlihat walaupun tidak dapat dimengerti, sedangkan pada steganografi, kecurigaan tersebut akan hilang karena data yang diamankan akan disembunyikan kedalam data lain yang sifatnya tidak rahasia.

Di samping itu dengan adanya pasal 41 UU no. 36 tahun 1999 tentang Telekomunikasi yang berbunyi “Dalam rangka pembuktian kebenaran pemakaian fasilitas telekomunikasi atas permintaan pengguna jasa telekomunikasi, penyelenggara jasa telekomunikasi wajib melakukan perekaman pemakaian fasilitas telekomunikasi yang digunakan oleh pengguna jasa telekomunikasi dan dapat melakukan perekaman informasi sesuai dengan peraturan perundang-undangan yang berlaku”. Dari uraian diatas penulis merasa bahwa perlu dilakukan tindak pencegahan pencurian data dengan membuat aplikasi yang dapat mengamankan data. Dalam hal ini adalah pesan suara. Olehnya itu akan dirancang aplikasi yang dapat mengamankan

pesan suara dengan menggunakan kedua metode diatas, yaitu kriptografi dan steganografi.

### **B. Rumusan Masalah**

Berdasarkan latar belakang yang telah dijabarkan, maka pokok permasalahan yang dihadapi adalah bagaimana cara merancang dan membuat aplikasi steganografi penyembunyi pesan suara terenkripsi berbasis Android?

### **C. Fokus Penelitian dan Deskripsi Fokus**

#### **1. Fokus Penelitian**

Agar dalam pengerjaan tugas akhir lebih terarah, maka fokus penelitian pada penulisan ini fokus pada pembahasan berikut :

1. Aplikasi ini berjalan pada *smartphone* berbasis Android minimal versi 4.4 Kitkat.
2. Aplikasi ini menggunakan algoritma *serpent* untuk proses enkripsi dan dekripsinya.
3. Aplikasi ini menggunakan metode EOF (*end of file*) pada proses steganografinya.
4. Inputan harus berupa suara berekstensi .wav dan gambar berekstensi .jpg.
5. Aplikasi ini tidak bersifat *real-time*

## 2. Deskripsi Fokus

Agar tidak terjadi kesalahpahaman antara penulis dan pembaca, maka dikemukakan penjelasan yang sesuai dengan deskripsi fokus dalam penelitian ini. Adapun deskripsi fokus pada penelitian ini sebagai berikut :

1. Aplikasi ini berjalan di Android karena penulis akan membuat aplikasi steganografi berbasis mobile dan banyaknya pengguna Android di Indonesia.
2. Aplikasi ini menggunakan kriptografi algoritma *serpent* karena algoritma *serpent* lebih aman dibandingkan AES *Rijndael* walaupun *serpent* lebih lambat dibandingkan AES *Rijndael*.
3. Metode steganografi EOF (*end of file*) digunakan pada aplikasi ini karena metode ini mudah diterapkan pada bahasa pemrograman.
4. Inputan suara menggunakan ekstensi .wav yang merupakan ekstensi suara hasil rekaman. Dan inputan gambar menggunakan ekstensi .jpg karena merupakan format gambar yang paling umum digunakan.
5. Aplikasi ini tidak bersifat real-time karena proses enkripsi ataupun dekripsi tidak bergantung pada batasan waktu.

## D. Kajian Pustaka

Ada banyak aplikasi keamanan yang telah dibuat, baik itu kriptografi ataupun steganografi. Namun pada proses penelitiannya metode dan media yang digunakan berbeda, bahkan ada pula objeknya yang berbeda, berikut beberapa aplikasi enkripsi dan dekripsi yang telah ada :

Nugroho (2010), merancang aplikasi enkripsi SMS berbasis J2ME dengan menggunakan metode *vigenere cipher*. Aplikasi ini mengamankan pesan singkat yang akan dikirim dengan mengenkripsi isi pesannya dengan basis J2ME.

Perbedaan dengan penelitian yang penulis lakukan adalah pada sistem operasi yang digunakan dan metode algoritma yang digunakan, meskipun sama-sama berjalan pada telepon seluler. Pada penelitian yang dilakukan Nugroho, menggunakan sistem operasi J2ME, sedangkan penelitian yang dilakukan oleh penulis menggunakan sistem operasi Android, yang sudah tidak asing didengar.

Kemudian jenis algoritma enkripsi yang digunakan ialah *vigenere cipher*, sedangkan penulis menggunakan algoritma *serpent* pada penelitian ini. Data yang dienkripsi hanyalah sebatas teks, sedangkan penulis melakukan enkripsi terhadap suara dan kemudian melakukan teknik steganografi untuk proses penyembunyian datanya. Persamaannya sendiri yaitu sama-sama melakukan teknik pengamanan kriptografi untuk mengamankan pesannya dan berjalan pada telepon seluler.

Irawan dan Adriantantri (2010), merancang sebuah teknologi keamanan steganografi dengan judul “Steganografi untuk Menyembunyikan Suara dengan Smart Card sebagai Kunci Enkripsi”. Pada penelitian yang dilakukan oleh Irawan dan Adriantantri, aplikasi yang dirancang bertujuan untuk menyembunyikan pesan suara ke media gambar, dengan terlebih dahulu suara dienkripsi pada *smartcard*.

Secara garis besar, penelitian yang dilakukan Irawan dan Adriantantri sama dengan yang dilakukan penulis yaitu melakukan enkripsi suara dan menyembunyikannya ke dalam gambar sebagai media penyembunyiannya,

perbedaannya pada penelitian Irawan dan Adriantantri menggunakan *smartcard* yang telah siap pakai untuk proses enkripsinya.

Putri (2009), merancang aplikasi enkripsi dan dekripsi suara dengan judul “Studi dan Implementasi Enkripsi Pengiriman Suara dengan Algoritma *Serpent*”. Pada aplikasi yang dirancang oleh Putri, suara yang dikirimkan akan dikonversi ke bit terlebih dahulu, kemudian dienkripsi sebelum dikirim ke tujuan, sehingga suara yang dikirim keamanan datanya terjamin dari orang-orang yang akan melakukan penyadapan, dan kemudian akan didekripsi setelah sampai ke tujuan lalu dikonversi menjadi suara.

Perbedaan pada penelitian yang dilakukan oleh Putri dan penulis lakukan ialah media yang digunakan oleh Putri adalah PC, dengan *file* yang akan dikirim berupa *file* suara dan juga dapat langsung melalui *microphone*, sedangkan pada penelitian yang akan penulis lakukan menggunakan media *smartphone* berbasis Android dan melakukan teknik steganografi. Persamaannya sendiri yaitu sama-sama melakukan enkripsi pada *file audio* dengan menggunakan algoritma yang sama, yaitu algoritma *serpent*.

Ayuningsih (2014), merancang aplikasi steganografi dengan judul “aplikasi penyembunyi pesan pada *file* citra dengan menggunakan metode steganografi *eof* (*end of file*) berbasis *mobile*”. Pada aplikasi yang dirancang oleh Ayuningsih, pesan yang berupa teks akan ditanamkan kedalam *file* gambar, dengan terlebih dahulu, teks dirubah ke dalam bentuk bit.

Pada penelitian yang dilakukan oleh Ayuningsih, data yang ditanamkan ke dalam gambar berupa teks, dan kemudian akan dikirim ke penerima. Sedangkan pada penelitian yang dilakukan penulis akan menanamkan *file* suara ke dalam gambar, kemudian dikirim ke penerima. Persamaan antara penelitian yang dilakukan oleh Ayuningsih dan yang dilakukan oleh peneliti adalah sama-sama melakukan teknik pengamananan Steganografi, dengan menggunakan metode yang sama, yaitu *end of fie*.

#### ***E. Tujuan dan Kegunaan Penelitian***

##### **1. Tujuan Penelitian**

Adapun tujuan penelitian ini adalah untuk merancang dan membuat aplikasi steganografi menyembunyi pesan suara terenkripsi berbasis android yang nantinya menghasilkan suatu aplikasi steganografi yang berfungsi untuk mengamankan pesan suara yang telah terenkripsi dan tersembunyi di dalam format gambar.

##### **2. Kegunaan Penelitian**

###### **a. Kegunaan bagi dunia akademik**

Sebagai kontribusi ilmu di bidang teknologi informasi dan untuk pengembangan pada masa ke depannya.

###### **b. Kegunaan bagi masyarakat**

Sebagai layanan penyembunyian data suara ke dalam gambar.

**c. Kegunaan bagi penulis**

Dapat menerapkan ilmu teknologi informasi yang didapatkan dari bangku perkuliahan maupun secara otodidak, menambah wawasan penulis dalam bidang keamanan data serta sebagai salah satu syarat untuk meraih gelar sarjana.



## BAB II

### TINJAUAN TEORETIS

#### A. Aplikasi

Aplikasi adalah suatu perangkat lunak atau software siap pakai yang berjalan di atas sistem operasi, dimana aplikasi membutuhkan interaksi antara manusia dan komputer. Istilah aplikasi berasal dari bahasa Inggris “*application*” yang berarti penerapan, lamaran ataupun penggunaan. Sedangkan secara istilah, pengertian aplikasi adalah suatu program yang siap untuk digunakan yang dibuat untuk melaksanakan suatu fungsi bagi pengguna jasa aplikasi serta penggunaan aplikasi lain yang dapat digunakan oleh suatu sasaran yang akan dituju. Menurut kamus komputer eksekutif, aplikasi mempunyai arti yaitu pemecahan masalah yang menggunakan salah satu teknik pemrosesan data aplikasi yang biasanya berpacu pada sebuah komputasi yang diinginkan atau diharapkan maupun pemrosesan data yang diharapkan. (Lepank, 2012)

Adapun beberapa pendapat menurut para ahli tentang aplikasi :

1. Ali Zaki & *SMITDEV COMMUNITY*

Aplikasi adalah komponen yang berguna melakukan pengolahan data maupun kegiatan-kegiatan seperti pembuatan dokumen atau pengolahan data. Aplikasi adalah bagian *PC* yang berinteraksi langsung dengan *user*. Aplikasi berjalan di atas sistem operasi, sehingga agar aplikasi bisa diaktifkan, kita perlu melakukan instalasi sistem operasi terlebih dahulu.



## 2. Hengky W. Pramana

Aplikasi adalah satu unit perangkat lunak yang dibuat untuk melayani kebutuhan akan beberapa aktivitas seperti sistem perniagaan, *game*, pelayanan masyarakat, periklanan, atau semua proses yang hampir dilakukan manusia

## 3. Harip Santoso

Aplikasi adalah suatu kelompok *file* (*form*, *class*, *report*) yang bertujuan untuk melakukan aktivitas tertentu yang saling terkait, misalnya aplikasi *payroll*, aplikasi *fixed asset*, dll

## 4. Yuhefizar

Aplikasi merupakan program yang dikembangkan untuk memenuhi kebutuhan pengguna dalam menjalankan pekerjaan tertentu.

## 5. Rachmad Hakim. S

Aplikasi merupakan perangkat lunak yang digunakan untuk tujuan tertentu, seperti mengolah dokumen, mengatur *Windows* &, permainan, dan sebagainya.

## 6. R. Eko. I & Djokopran

Aplikasi merupakan proses atau prosedur aliran data dalam infrastruktur teknologi informasi yang dapat dimanfaatkan oleh para pengambil keputusan yang sesuai dengan jenjang dan kebutuhan (relevan)

## **B. Keamanan**

Keamanan adalah keadaan bebas bahaya. Istilah ini bisa digunakan dengan hubungan kepada kejahatan, segala bentuk kecelakaan, dan lain-lain. Keamanan merupakan topik yang luas termasuk keamanan nasional terhadap serangan teroris, keamanan komputer terhadap *hacker*, keamanan rumah terhadap maling dan penyusup lainnya, keamanan finansial terhadap kehancuran ekonomi dan banyak situasi berhubungan lainnya. Kemudian ada yang namanya keamanan jaringan, Keamanan jaringan adalah kumpulan piranti yang dirancang untuk melindungi data ketika transmisi terhadap ancaman pengaksesan, pengubahan dan penghalangan oleh pihak yang tidak berwenang. Berbeda dengan keamanan komputer, merupakan kumpulan piranti yang dirancang untuk melindungi komputer sehingga data pada komputer terlindungi. Sistem anti virus komputer merupakan contoh piranti keamanan komputer sedangkan protokol web yang aman merupakan contoh peranti keamanan jaringan. (Sadikin, 2012: 1)

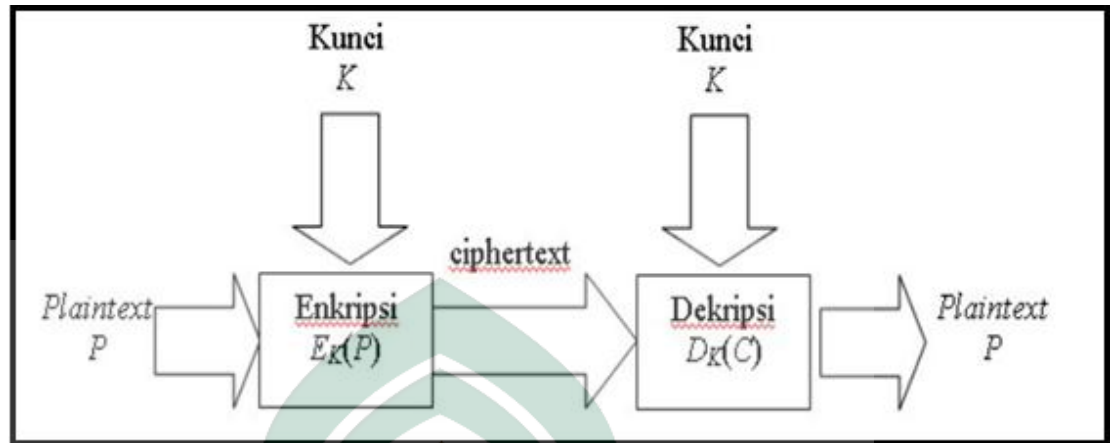
Berikut beberapa metode yang digunakan dalam mengamankan data :

### **1. Kriptografi**

Kriptografi (*cryptography*) berasal dari bahasa Yunani, terdiri dari dua suku kata yaitu *kripto* dan *graphia*. *Kripto* artinya menyembunyikan, sedangkan *graphia* artinya tulisan. Kriptografi pada awalnya dijabarkan sebagai ilmu yang mempelajari bagaimana menyembunyikan pesan. Namun pada pengertian modern kriptografi adalah ilmu yang bersandarkan pada teknik matematika untuk

berurusan dengan keamanan informasi seperti kerahasiaan, keutuhan data dan otentikasi entitas. Jadi pengertian kriptografi modern adalah tidak saja urusan hanya dengan penyembunyian pesan namun lebih pada sekumpulan teknik yang menyediakan keamanan informasi. (Sadikin, 2012:9)

Kriptografi memiliki 4 komponen utama yaitu, *plaintext*, yaitu pesan yang dapat dibaca, *ciphertext*, yaitu pesan acak yang tidak dapat dibaca, *key*, yaitu kunci untuk melakukan teknik kriptografi, Algoritma, yaitu metode untuk melakukan enkripsi dan dekripsi. Enkripsi adalah proses merubah *plainteks* menjadi *cipherteks* atau proses mengubah data yang dimengerti menjadi tidak dimengerti. Dan dekripsi adalah proses mengembalikan *cipherteks* menjadi *plainteks*. Terdapat dua algoritma kriptografi berdasarkan kesamaan kuncinya yaitu, kriptografi kunci simetris dan kriptografi dan asimetris. Kriptografi kunci simetris hanya menggunakan satu *key*/kunci untuk melakukan enkripsi dan dekripsi data secara keseluruhan, sedangkan kriptografi kunci asimetris adalah algoritma yang menggunakan kunci yang berbeda untuk proses enkripsi dan dekripsinya. Skema ini disebut juga sebagai sistem kriptografi kunci publik karena kunci untuk enkripsi dibuat untuk diketahui oleh umum (*public-key*) atau dapat diketahui siapa saja, tapi untuk proses dekripsinya hanya dapat dilakukan oleh yang berwenang yang memiliki kunci rahasia untuk mendekripsinya, disebut *private-key*. Secara umum proses kriptografi seperti berikut :



Gambar II.1 : Proses Kriptografi (Arif, 2014)

Berikut contoh pada proses kriptografi secara umum :

- a. *Plaintext* : “kita bertemu di bandara Soekarno Hatta pada pukul 12.30”

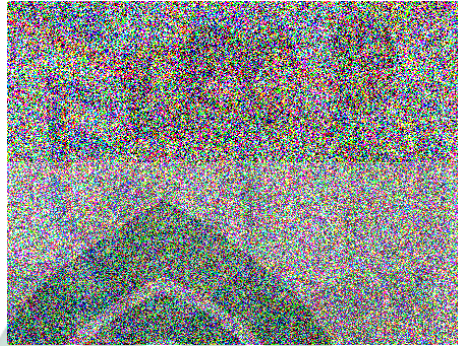
*Ciphertext* : “asdfua adfargafad ghteryds dfgy34 sdfgdfhhhs hfdsh”

- b. *Plaintext*



Gambar II.2 : Gambar yang belum dienkripsi (Merdeka, 2013)

### *Ciphertext*



Gambar II.3 : Gambar setelah dienkripsi (Merdeka, 2013)

Ada banyak jenis algoritma kriptografi, salah satunya adalah algoritma *serpent*. *Serpent* merupakan algoritma kriptografi yang bersifat *block chipper symmetric cryptography* yang merupakan AES finalis pada kontes AES, dimana kontes tersebut dimenangkan *Rijndael* AES. *Serpent* ditemukan oleh Ross Anderson, Eli Biham dan Lars Knudsen.

Seperti halnya AES yang lain, *serpent* mempunyai ukuran blok sebesar 128 *bit* dan dapat mendukung key dengan ukuran 128 *bit*, 192 *bit*, ataupun 256 *bit*. *Serpent* mengimplementasikan kriptopsistem 32 tahap *substitution-permutation network*, dimana tahap-tahapan tersebut megoperasikan 4 buah variable dengan ukuran 32 bit. Masing-masing tahap menggunakan delapan buah 4 bit x 4 bit S-Box secara Paralel.

## 2. Steganografi

Steganografi adalah seni dan ilmu menulis pesan tersembunyi atau menyembunyikan pesan dengan suatu cara sehingga selain pengirim dan

penerima, tidak ada seorangpun yang mengetahui atau menyadari bahwa ada suatu pesan rahasia.

Kini, istilah steganografi termasuk penyembunyian data digital dalam berkas (*file*) komputer. Contohnya, pengirim mulai dengan berkas gambar biasa, lalu mengatur warna setiap pixel ke-100 untuk menyesuaikan suatu huruf dalam alphabet (perubahannya begitu halus sehingga tidak ada seorangpun yang menyadarinya jika ia tidak benar-benar memperhatikannya).

Pada umumnya, pesan steganografi muncul dengan rupa lain seperti gambar, artikel, daftar belanjaan, atau pesan-pesan lainnya. Pesan yang tertulis ini merupakan tulisan yang menyelubungi atau menutupi. Contohnya, suatu pesan bisa disembunyikan dengan menggunakan tinta yang tidak terlihat di antara garis-garis yang kelihatan.

Teknik Steganografi meliputi banyak sekali metode komunikasi untuk menyembunyikan pesan rahasia (teks atau gambar) di dalam berkas-berkas lain yang mengandung teks, gambar, bahkan audio tanpa menunjukkan ciri-ciri perubahan yang nyata atau terlihat dalam kualitas dan struktur dari berkas semula. Metode ini termasuk tinta yang tidak tampak, *microdots*, pengaturan kata, tanda tangan digital, jalur tersembunyi dan komunikasi *spectrum* lebar.

Tujuan dari steganografi adalah merahasiakan atau menyembunyikan keberadaan dari sebuah pesan tersembunyi atau sebuah informasi. Dalam prakteknya, kebanyakan pesan disembunyikan dengan membuat perubahan tipis terhadap data digital lain yang isinya tidak akan menarik perhatian dari penyerang

potensial, sebagai contoh sebuah gambar yang terlihat tidak berbahaya. Perubahan ini bergantung pada kunci (sama pada kriptografi) dan pesan untuk disembunyikan. Orang yang menerima gambar kemudian dapat menyimpulkan informasi terselubung dengan cara mengganti kunci yang benar ke dalam algoritma yang digunakan.

Pada metode steganografi cara ini sangat berguna jika digunakan pada cara steganografi komputer karena banyak format berkas digital yang dapat dijadikan media untuk menyembunyikan pesan. Format yang biasa digunakan diantaranya, format *image* : bitmap (bmp, gif, pcx, jpeg, dll, format audio : waf, voc, mp3, dll, format lain : teks file, html, pdf, dll.

Kelebihan steganografi jika dibandingkan dengan kriptografi adalah pesan-pesannya tidak menarik perhatian orang lain. Pesan-pesan berkode dalam kriptografi yang tidak disembunyikan, walaupun tidak dapat dipecahkan, akan menimbulkan kecurigaan. Seringkali, steganografi dan kriptografi digunakan secara bersamaan untuk menjamin keamanan pesan rahasianya.

Sebuah pesan steganografi (*plaintext*), biasanya pertama-tama dienkripsikan dengan beberapa arti tradisional, yang menghasilkan *ciphertext*. Kemudian, *covertext* dimodifikasi dalam beberapa cara sehingga berisi *ciphertext*, yang menghasilkan *stegotext*. Contohnya, ukuran huruf, ukuran spasi, jenis huruf, atau karakteristik *covertext* lainnya dapat dimanipulasi untuk membawa pesan tersembunyi, hanya penerima (yang harus mengetahui teknik yang digunakan) dapat membuka pesan dan mendekripsinya.



Berikut beberapa metode dalam steganografi :

a. *Least Significant Bit Insertion (LSB)*

Metoda yang digunakan untuk menyembunyikan pesan pada media digital tersebut berbeda-beda. Contohnya, pada berkas *image* pesan dapat disembunyikan dengan menggunakan cara menyisipkannya pada bit rendah atau bit yang paling kanan (*LSB*) pada data *pixel* yang menyusun *file* tersebut. Pada berkas bitmap 24 bit, setiap *pixel* (titik) pada gambar tersebut terdiri dari susunan tiga warna merah, hijau dan biru (*RGB*) yang masing-masing disusun oleh bilangan 8 bit (*byte*) dari 0 sampai 255 atau dengan format biner 00000000 sampai 11111111. Dengan demikian, pada setiap *pixel* berkas bitmap 24 bit kita dapat menyisipkan 3 bit data.

Kekurangan dari *LSB Insertion*: Dapat diambil kesimpulan dari contoh 8 bit *pixel*, menggunakan *LSB Insertion* dapat secara drastis mengubah unsur pokok warna dari *pixel*. Ini dapat menunjukkan perbedaan yang nyata dari *cover image* menjadi *stego image*, sehingga tanda tersebut menunjukkan keadaan dari steganografi. Variasi warna kurang jelas dengan 24 bit *image*, bagaimanapun *file* tersebut sangatlah besar. Antara 8 bit dan 24 bit *image* mudah diserang dalam pemrosesan *image*, seperti *cropping* (kegagalan) dan *compression* (pemampatan).

Keuntungan dari *LSB Insertion* : Keuntungan yang paling besar dari algoritma *LSB* ini adalah cepat dan mudah. Dan juga algoritma tersebut memiliki *software* steganografi yang mendukung dengan bekerja di antara unsur pokok warna *LSB* melalui manipulasi *pallette* (lukisan).



### b. *Algorithms and Transformation*

Algoritma *compression* adalah metode steganografi dengan menyembunyikan data dalam fungsi matematika. Dua fungsi tersebut adalah *Discrete Cosine Transformation* (DCT) dan *Wavelet Transformation*. Fungsi DCT dan Wavelet yaitu mentransformasi data dari satu tempat (domain) ke tempat (domain) yang lain. Fungsi DCT yaitu mentransformasi data dari tempat spatial (*spatial domain*) ke tempat frekuensi (*frequency domain*).

### c. *Redundant Pattern Encoding*

*Redundant Pattern Encoding* adalah menggambar pesan kecil pada kebanyakan gambar. Keuntungan dari metode ini adalah dapat bertahan dari *cropping* (kegagalan). Kerugiannya yaitu tidak dapat menggambar pesan yang lebih besar.

### d. *Spread Spectrum method*

*Spread Spectrum* steganografi terpencah-pencar sebagai pesan yang diacak (*encrypted*) melalui gambar (tidak seperti dalam LSB). Untuk membaca suatu pesan, penerima memerlukan algoritma yaitu *crypto-key* dan *stego-key*. Metode ini juga masih mudah diserang yaitu penghancuran atau pengrusakan dari kompresi dan proses *image* (gambar). (Wikipedia, 2014)

### e. *End Of File*

Teknik yang digunakan pada *steganografi* beragam tetapi secara umum teknik ini menggunakan *redundant bits* sebagai tempat menyembunyikan pesan pada saat dilakukan kompresi data, dan kemudian menggunakan kelemahan indera manusia

yang tidak sensitive sehingga pesan tersebut tidak ada perbedaan yang terlihat atau yang terdengar. Teknik EOF atau *End of File* merupakan salah satu teknik yang digunakan dalam steganografi. Teknik ini digunakan dengan cara menambahkan data atau pesan rahasia pada akhir *file*. Teknik ini dapat digunakan untuk menambahkan data yang ukurannya sesuai dengan kebutuhan. Perhitungan kasar ukuran *file* yang telah disisipkan data sama dengan ukuran *file* sebelum disisipkan data ditambah ukuran data rahasia yang telah diubah menjadi *encoding file*.

Seperti Kriptografi dan Kriptanalisis, Steganalisis didefinisikan sebagai suatu seni dan ilmu dalam mendeteksi informasi tersembunyi. Sebagai tujuan dari steganografi adalah untuk merahasiakan keberadaan dari sebuah pesan rahasia, satu keberhasilan penyerangan pada sebuah sistem steganografi terdiri dari pendeteksian bahwa sebuah berkas yang diyakini berisikan data terselubung. Seperti dalam Kriptanalisis, diasumsikan bahwa sistem steganografi telah diketahui oleh si penyerang. Maka dari itu, keamanan dari sistem steganografi bergantung hanya pada fakta bahwa kunci rahasia tidak diketahui oleh si penyerang.

*Stegosystem* di sini berisi tentang penyerangan-penyerangan yang dilakukan terhadap suatu sistem steganografi, sebuah perbedaan penting harus dibuat di antara penyerangan-penyerangan pasif di mana penyerang hanya dapat memotong data, dan penyerangan-penyerangan aktif di mana penyerang juga dapat memanipulasi data. Gambar di bawah ini menunjukkan sebuah diagram untuk menjelaskan sistem stego. Lingkaran-lingkaran menunjukkan tempat-tempat penyerang yang berpotensi

memiliki jalan masuk ke satu atau lebih dari tempat-tempat tersebut akibat penyerangan-penyerangan yang berbeda jenis, dan juga berfungsi untuk melakukan sebuah penyerangan aktif. Jika lingkaran tidak terisi, penyerang hanya dapat melakukan penyerangan pasif yaitu menghalangi memotong data. (Wikipedia, 2014)

Penyerangan-penyerangan berikut memungkinkan dalam model dari stegosistem ini:

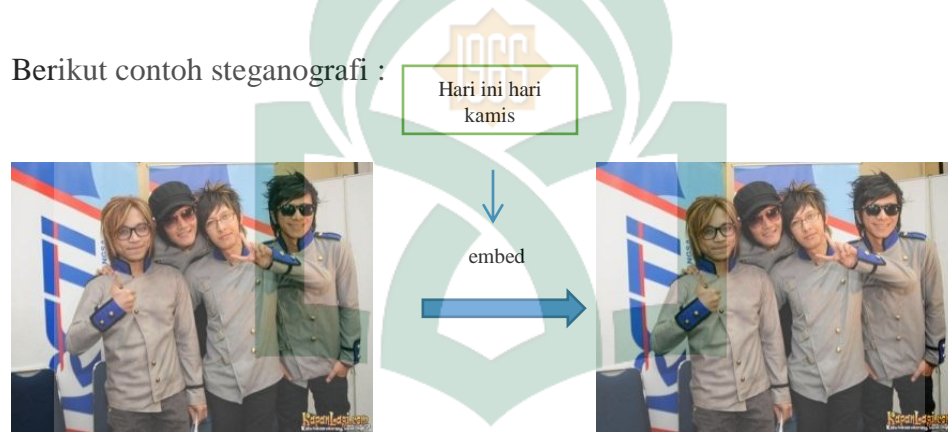
- a. *Stego-Only-Attack* (Penyerangan hanya Stego). Penyerang telah menghalangi stego data dan dapat menganalisisnya.
- b. *Stego-Attack* (Penyerangan Stego). Pengirim telah menggunakan *cover* yang sama berulang kali untuk data terselubung. Penyerang memiliki berkas stego yang berasal dari *cover file* yang sama. Dalam setiap berkas stego tersebut, sebuah pesan berbeda disembunyikan.
- c. *Cover-Stego-Attack* (Penyerangan selubung Stego). Penyerang telah menghalangi berkas stego dan mengetahui *cover file* mana yang digunakan untuk menghasilkan berkas stego ini. Ini menyediakan sebuah keuntungan melalui penyerangan *stego-only* untuk si penyerang.
- d. *Manipulating the stego data* (Memanipulasi data stego). Penyerang memiliki kemampuan untuk memanipulasi data stego. Jika penyerang hanya ingin menentukan sebuah pesan disembunyikan dalam berkas stego ini, biasanya ini tidak memberikan sebuah keuntungan, tapi memiliki kemampuan

dalam memanipulasi data stego yang berarti bahwa si penyerang mampu memindahkan pesan rahasia dalam data stego (jika ada).

e. *Manipulating the cover data* (Memanipulasi data terselubung).

Penyerang dapat memanipulasi data terselubung dan menghalangi hasil data stego. Ini dapat membuat tugas dalam menentukan apakah data stego berisikan sebuah pesan rahasia lebih mudah bagi si penyerang.

Berikut contoh steganografi :



Gambar II.4 : Contoh Steganografi (Merdeka, 2013)

### C. Android

#### 1. Sejarah Android

Android adalah sistem operasi berbasis Linux yang dirancang untuk perangkat seluler layar sentuh seperti *smartphone* dan komputer tablet. Android awalnya dikembangkan oleh Android, Inc., dengan dukungan finansial dari Google, yang kemudian membelinya pada tahun 2005. Sistem operasi ini dirilis secara resmi pada tahun 2007, bersamaan dengan didirikannya *Open Handset Alliance*, konsorsium dari perusahaan-perusahaan perangkat keras,

perangkat lunak, dan telekomunikasi yang bertujuan untuk memajukan standar terbuka perangkat seluler. Ponsel Android pertama mulai dijual pada bulan Oktober 2008. (Wikipedia, 2014)

## 2. Versi Android

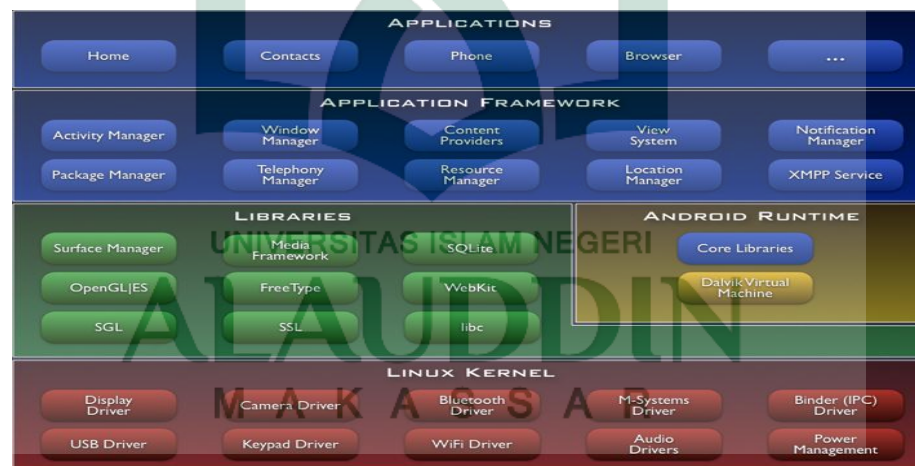
Versi Android diawali dengan dirilisnya Android beta pada bulan November 2007. Versi komersial pertama, Android 1.0, dirilis pada September 2008. Android dikembangkan secara berkelanjutan oleh Google dan *Open Handset Alliance* (OHA), yang telah merilis sejumlah pembaruan sistem operasi ini sejak dirilisnya versi awal.

Sejak April 2009, versi Android dikembangkan dengan nama kode yang dinamai berdasarkan makanan pencuci mulut dan penganan manis. Masing-masing versi dirilis sesuai urutan alfabet, yakni *Cupcake* (1.5), *Donut* (1.6), *Eclair* (2.0–2.1), *Froyo* (2.2–2.2.3), *Gingerbread* (2.3–2.3.7), *Honeycomb* (3.0–3.2.6), *Ice Cream Sandwich* (4.0–4.0.4), *Jelly Bean* (4.1–4.3), dan *KitKat* (4.4+). Pada tanggal 3 September 2013, Google mengumumkan bahwa sekitar 1 miliar perangkat seluler aktif di seluruh dunia menggunakan OS Android. (Wikipedia, 2014)

## 3. Arsitektur Sistem Operasi Android

Google sebagai pencipta Android yang kemudian diasuh oleh *Open Handset Alliance* mengibaratkan Android sebagai sebuah tumpukan software. Setiap lapisan dari tumpukan ini menghimpun beberapa program yang

mendukung fungsi-fungsi spesifik dari sistem operasi. Tumpukan paling bawah adalah kernel. Google menggunakan kernel Linux versi 2.6 untuk membangun Android, yang mencakup *memory management*, *security setting*, *power management*, dan beberapa *driver hardware*. Bertempat di level yang sama dengan library adalah lapisan *runtime* yang mencakup serangkaian inti *library* Java. Dengannya, para programmer dapat mengembangkan aplikasi untuk Android menggunakan bahasa pemrograman Java. Lapisan selanjutnya adalah *application framework*, yang mencakup program untuk mengatur fungsi - fungsi dasar *smartphone*. (Dony, 2011)



Gambar II.5 : Arsitektus OS Android (Donny, 2011)

Adapun penjelasan dari gambar II.5 adalah sebagai berikut :

a. Linux Kernel

Android dibangun di atas kernel Linux 2.6. Namun secara keseluruhan android bukanlah linux, karena dalam android tidak terdapat paket standar yang

dimiliki oleh linux lainnya. Linux merupakan sistem operasi terbuka yang handal dalam manajemen memori dan proses. Oleh karenanya pada android hanya terdapat beberapa servis yang diperlukan seperti keamanan, manajemen memori, manajemen proses, jaringan dan driver. Kernel linux menyediakan driver layar, kamera, keypad, WiFi, Flash Memory, audio, dan IPC (*Interprocess Communication*) untuk mengatur aplikasi dan lubang keamanan.

#### *b. Libraries*

Android menggunakan beberapa paket pustaka yang terdapat pada C/C++ dengan standar *Berkeley Software Distribution* (BSD) hanya setengah dari yang aslinya untuk tertanam pada kernel Linux. Beberapa pustaka diantaranya:

- 1) *Media Library* untuk memutar dan merekam berbagai macam format audio dan video.
- 2) *Surface Manager* untuk mengatur hak akses layer dari berbagai aplikasi.
- 3) *Graphic Library* termasuk didalamnya SGL dan OpenGL, untuk tampilan 2D dan 3D.
- 4) *SQLite* untuk mengatur relasi database yang digunakan pada aplikasi.
- 5) *SSL* dan *WebKit* untuk browser dan keamanan internet.

#### *c. Android Runtime*

Pada android tertanam paket pustaka inti yang menyediakan sebagian besar fungsi android. Inilah yang membedakan Android dibandingkan dengan sistem operasi lain yang juga mengimplementasikan Linux. Android Runtime merupakan

mesin virtual yang membuat aplikasi android menjadi lebih tangguh dengan paket pustaka yang telah ada.

*d. Application Framework*

Kerangka aplikasi menyediakan kelas-kelas yang dapat digunakan untuk mengembangkan aplikasi android. Selain itu, juga menyediakan abstraksi generik untuk mengakses perangkat, serta mengatur tampilan user interface dan sumber daya aplikasi. Bagian terpenting dalam kerangka aplikasi android adalah sebagai berikut

- 1) *Activity Manager*, berfungsi untuk mengontrol siklus hidup aplikasi dan menjaga keadaan ”*Backstack*“ untuk navigasi penggunaan.
  - 2) *Content Providers*, berfungsi untuk merangkum data yang memungkinkan digunakan oleh aplikasi lainnya, seperti daftar nama.
  - 3) *Resource Manager*, untuk mengatur sumber daya yang ada dalam program. Serta menyediakan akses sumber daya di luar kode program, seperti karakter, grafik, dan file *layout*.
  - 4) *Location Manager*, berfungsi untuk memberikan informasi detail mengenai lokasi perangkat Android berada.
  - 5) *Notification Manager*, mencakup berbagai macam peringatan seperti, pesan masuk, janji, dan lain sebagainya yang akan ditampilkan pada status bar.
- (Donny, 2011)

*e. Application Layer*

Puncak dari diagram arsitektur android adalah lapisan aplikasi dan widget. Lapisan aplikasi merupakan lapisan yang paling tampak pada pengguna ketika

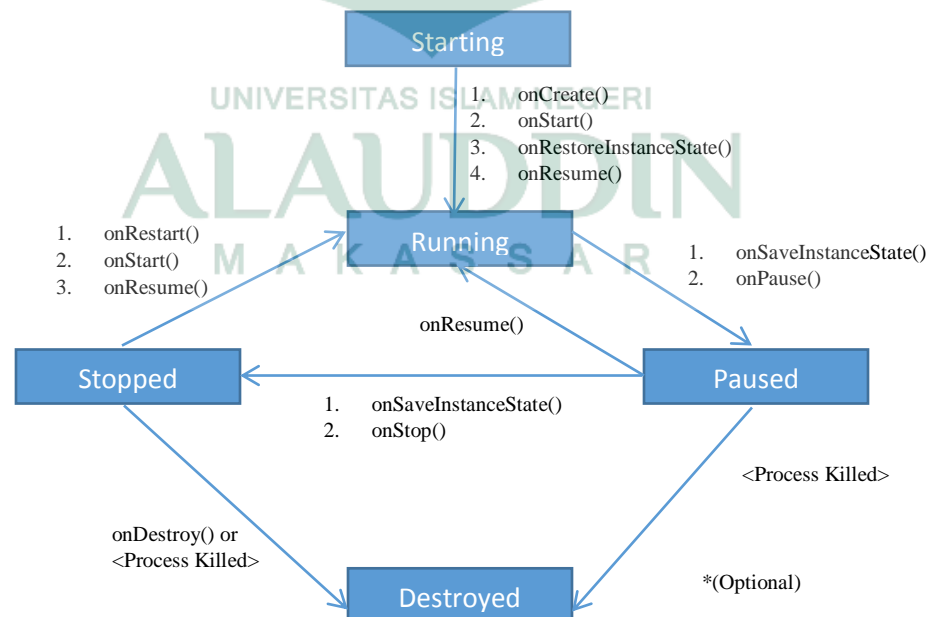


menjalankan program. Pengguna hanya akan melihat program ketika digunakan tanpa mengetahui proses yang terjadi dibalik lapisan aplikasi. Lapisan ini berjalan dalam Android *runtime* dengan menggunakan kelas dan *service* yang tersedia pada *framework* aplikasi.

Lapisan aplikasi Android sangat berbeda dibandingkan dengan sistem operasi lainnya. Pada Android semua aplikasi, baik aplikasi inti (*native*) maupun aplikasi pihak ketiga berjalan diatas lapisan aplikasi dengan menggunakan pustaka API (*Application Programming Interface*) yang sama.

#### 4. Siklus Activity Android

Berikut adalah gambar siklus activity android :



Gambar II.6 : Siklus Activity Android

Selama siklus ini berjalan, activity bisa mempunyai lebih dari 2 status seperti yang terlihat pada gambar II.6. setiap status tidak bisa dikontrol karena semuanya sudah ditangani oleh sistem. Namun akan menampilkan pesan saat terjadi perubahan status melalui method `onXX()`. (Huda, 2012:10).

Adapun penjelasan setiap status dari siklus activity android di atas adalah :

*a. onCreate(Bundle)*

Dipanggil saat pertama kali aplikasi dijalankan. Dapat digunakan untuk deklarasi variabel atau membuat *user interface*.

*b. onStart()*

Mengindikasikan *activity* yang ditampilkan ke pengguna (user). *onResume()* dipanggil saat aplikasi mulai berinteraksi dengan pengguna. Disini sangat cocok untuk meletakkan animasi ataupun musik.

*c. onPause()*

Dipanggil saat aplikasi yang dijalankan kembali ke halaman sebelumnya atau biasanya karena ada *activity* baru yang dijalankan. Disini cocok untuk meletakkan algoritma penyimpanan (save). *onStop()* Dipanggil saat aplikasi berjalan di belakang layar dalam waktu cukup lama.

*d. onRestart()*

Activity kembali menampilkan user interface setelah status stop.

*e. onDestroy()*

Dipanggil saat aplikasi benar-benar berhenti.

*f. onSaveInstanceState(Bundle)*

Method ini mengijinkan *activity* untuk menyimpan setiap status *intance*.

Misalnya dalam mengedit teks, kursor bergerak dari kiri ke kanan.

g. *onRestoreInstanceState(Bundle)*





Dipanggil saat activity kembali meng-inisialisasi dari status sebelumnya yang disimpan oleh *onSaveInstanceState(Bundle)*. (Huda , 2012)




## 5. Daftar Simbol

### a. Daftar Simbol *Flowmap Diagram*

*Flowmap* atau bagan alir adalah bagan yang menunjukkan aliran di dalam program atau prosedur sistem secara logika. *Flowmap* ini berfungsi untuk memodelkan masukan, keluaran, proses maupun transaksi dengan menggunakan simbol-simbol tertentu. Pembuatan *flowmap* ini harus dapat memudahkan bagi pemakai dalam memahami alur dari sistem atau transaksi.

Tabel II.1 Daftar Simbol Flowmap Diagram (Jogiyanto, 2001)

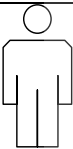


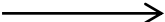
Simbol	Nama	Keterangan
	Terminator awal / akhir program	Simbol untuk memulai dan mengakhiri suatu program
	Dokumen	Menunjukkan dokumen berupa dokumen input dan output pada proses manual dan proses berbasis computer
	Proses Manual	Menunjukkan kegiatan proses yang dilakukan secara manual.
	Proses Komputer	Menunjukkan kegiatan proses yang dilakukan secara komputerisasi

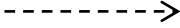

	Arah aliran data	Menunjukkan arah aliran dokumen antar bagian yang terkait pada suatu sistem.
	Penyimpanan Manual	Menunjukkan media penyimpanan data / informasi secara manual
	Data	Simbol input/output digunakan untuk mewakili data input/output

b. Daftar Simbol *Use Case Diagram*

*Use case diagram* merupakan gambaran scenario dari interaksi antara pengguna dengan sistem. *Use case diagram* menggambarkan hubungan antara aktor dan kegiatan yang dapat dilakukannya terhadap aplikasi.

Tabel II.2 Daftar Simbol *Use Case Diagram* (Booch, 1999)



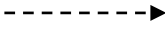

Simbol	Nama	Keterangan
	<i>Actor</i>	Menspesifikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
	<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil terukur bagi suatu actor.
	System	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
	<i>Unidirectional Association</i>	Menggambarkan relasi antara <i>actor</i> dengan <i>use case</i> dan proses berbasis computer.

	<i>Dependencies or Instantiates</i>	Menggambarkan kebergantungan ( <i>dependencies</i> ) antar <i>item</i> dalam diagram
	<i>Generalization</i>	Menggambarkan relasi lanjut antar <i>use case</i> atau menggambarkan struktur pewarisan antar <i>actor</i>

c. Daftar Simbol *Class Diagram*

*Class Diagram* merupakan diagram yang menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem.



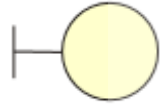

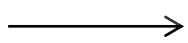
Tabel II.3 Daftar Simbol *Class Diagram* (Booch, 1999)

Simbol	Nama	Keterangan
<div style="border: 1px solid black; padding: 5px; width: fit-content;"> <p>Nama Class</p> <p>+ atribut</p> <p>+ atribut</p> <p>+ atribut</p> <hr/> <p>+ method</p> <p>+ method</p> </div>	Class	Blok - blok pembangun pada pemrograman berorientasi obyek. Terdiri atas 3 bagian. Bagian atas adalah bagian nama dari class. Bagian tengah mendefinisikan property/atribut class. Bagian akhir mendefinisikan method- method dari sebuah class.
	Association	Menggambarkan relasi asosiasi
	Composition	Menggambarkan relasi komposisi
	Dependencies	Menggambarkan relasi dependensi
	Aggregation	Menggambarkan relasi agregat

d. Daftar Simbol *Sequence Diagram*

*Sequence diagram* menggambarkan interaksi antar objek di dalam dan di sekitar sistem berupa *message* terhadap waktu. Pembuatan *sequence diagram* bertujuan agar perancangan aplikasi lebih mudah dan terarah.





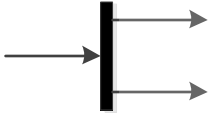
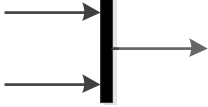
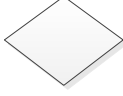
Tabel II.4 Daftar Simbol *Sequence Diagram* (Rosenberg, 2007)

Simbol	Nama	Keterangan
	<i>Actor</i>	Menggambarkan actor pada <i>sequence diagram</i>
	<i>Control</i>	Menggambarkan unsur kendali pada diagram
	Boundary	Menggambarkan kelas batasan ( <i>boundary</i> ) pada diagram
	<i>Entity</i>	Menggambarkan kelas entitas pada diagram
	<i>Dependencies or Instantiates</i>	Menggambarkan pesan antar dua objek

e. Daftar Simbol *Activity Diagram*

*Activity Diagram* adalah representasi grafis dari seluruh tahapan alur kerja yang mengandung aktivitas, pilihan, tindakan, perulangan, dan hasil dari aktivitas tersebut. Diagram ini dapat digunakan untuk menjelaskan proses bisnis dan alur kerja operasional secara langkah demi langkah dari komponen suatu sistem.

Tabel II.5 Daftar Simbol *Activity Diagram* (Jogiyanto, 2001)

Simbol	Nama	Keterangan
	<i>Action</i>	State dari sistem yang mencerminkan eksekusi dari suatu aksi.
	<i>Start State</i>	Bagaimana objek dibentuk atau diawali
	<i>End State</i>	Bagaimana objek dibentuk atau diakhiri
	<i>State Transition</i>	Menunjukkan kegiatan apa berikutnya setelah suatu kegiatan
	<i>Fork</i>	Percabangan yang menunjukkan aliran pada activity diagram
	<i>Join</i>	Percabangan yang menjadi arah aliran pada activity diagram
	<i>Decision</i>	Pilihan untuk mengambil keputusan

### **BAB III**

## **METODELOGI PENELITIAN**

### ***A. Jenis dan Lokasi Penelitian***

Adapun jenis penelitian yang digunakan pada penelitian ini adalah penelitian lapangan kualitatif eksperimental. Penelitian kualitatif merupakan penelitian tentang riset yang bersifat deskriptif dan cenderung menggunakan analisis . Proses dan makna (perspektif subyek) lebih ditonjolkan dalam penelitian kualitatif. Landasan teori dimanfaatkan sebagai pemandu agar fokus penelitian sesuai dengan fakta di lapangan. Selain itu landasan teori juga bermanfaat untuk memberikan gambaran umum tentang latar penelitian dan sebagai bahan pembahasan hasil penelitian.. Tujuan penulis menggunakan metode ekperimental karena akan dilakukan eksperimen terhadap variable-variabel input untuk menganalisis *output* yang dihasilkan. Pada penelitian ini, lokasi penelitian dilakukan di Laboratorium Pemrograman Dasar Jurusan Teknik Informatika, Fakultas Sains dan Teknologi UIN Alauddin Makassar.

### ***B. Pendekatan Penelitian***

Adapun pendekatan penelitian yang digunakan pada penelitian ini adalah pendekatan saintifik. Pendekatan saintifik yaitu pendekatan berdasarkan ilmu pengetahuan dan teknologi.



### ***C. Sumber Data***

Sumber data pada penelitian ini diperoleh dari kumpulan buku mengenai pemrograman android, keamanan data, *kriptografi* dan *steganografi* juga berasal dari e-book, jurnal tentang penelitian terdahulu yang berkaitan dengan penelitian yang dilakukan, serta dari dunia maya atau internet.

### ***D. Metode Pengumpulan Data***

Adapun metode pengumpulan data yang digunakan pada penelitian ini adalah studi literatur, studi dokumentasi dan observasi.

#### **1. Studi Literatur**

Studi literatur adalah metode pengumpulan data dengan cara mempelajari hal-hal yang berkaitan dengan penelitian yang dilakukan seperti membaca buku atau jurnal.

#### **2. Studi Dokumentasi**

Dengan mempelajari dokumentasi program serupa yang berupa source code untuk digunakan sebagai contoh dalam proses pembuatan perangkat lunak.

#### **3. Observasi**

Yaitu dengan mengamati proses secara langsung terhadap beberapa perangkat lunak yang sejenis.

Tujuan penulis menggunakan metode studi literatur yaitu untuk mengumpulkan referensi yang berkaitan dengan penelitian yang akan dilakukan.

### ***E. Instrumen Penelitian***

Adapun instrumen penelitian yang digunakan pada penelitian ini adalah:

#### **1. Perangkat Keras**

##### **a. Notebook Acer Aspire 4253 dengan spesifikasi sebagai berikut :**

*1) AMD Dual-Core Processor E-350*

*2) 3GB DDR3 @ 531MHz*

*3) 14.0" HD LED LCD*

*4) 500 GB HDD*

*5) DVD Super Multi DL drive*

##### **b. Smartphone Android Motorola Moto G dengan spesifikasi sebagai berikut :**

*1) Android OS, v4.4.4 Kitkat*

*2) CPU Quad-core 1.5 GHz Krait*

*3) GPU Adreno 320*

*4) 2G: EDGE/GPRS (850 / 900 / 1.800 / 1.900MHz), 3G: HSPA+21/5.76Mbps*

*5) GPS with A-GPS support and GLONASS*

*6) MicroUSB v2.0*

*7) 16M colors True HD IPS Plus capacitive touchscreen, 4.5" 720 x 1280 pixels*

#### **2. Perangkat Lunak**

*a. Eclipse Luna*

*b. SQLite Manager*

*c. ADT*

- d. JDK
- e. AVD
- f. *Windows 7 Ultimate 32-bit*
- g. *OS Android*

## ***F. Teknik Pengolahan dan Analisis Data***

### **1. Pengolahan Data**

Pengolahan data merupakan proses data yang diolah menjadi sesuatu yang lebih berguna, berupa informasi. Pada penelitian ini, metode pengolahan data yang akan digunakan adalah berikut : (Narbuko & Achmadi, 1999)

#### **a. Editing**

Yaitu proses memeriksa data yang telah terkumpul dengan tujuan mengurangi kesalahan dan kekurangan yang ada pada data.

#### **b. Koding**

Yaitu dengan mengklasifikasikan yang dilakukan dengan cara memberi tanda/kode pada data.

### **2. Analisis Data**

Teknik analisis data merupakan proses mengatur atau mengorganisasikan data dari hasil pengumpulan data yang dilakukan dengan tujuan mendapatkan hipotesa kerja yang akan dilakukan. Pada penelitian ini, penulis menggunakan analisis data kualitatif, yaitu analisis yang prosesnya berjalan terus-menerus sepanjang kegiatan lapangan dilakukan. Peneliti merumuskan hipotesis, dan

mencatat tema sepanjang studi tersebut. Sewaktu penelitian berjalan, beberapa hipotesis dihilangkan atau dibuang, atau dijernihkan, dan hipotesis baru dirumuskan untuk diuji dengan data yang ada. (Tiro, 2011:122)

#### ***G. Metode Perancangan Aplikasi***

Pada penelitian ini, metode perancangan aplikasi yang digunakan adalah metode *prototype*. *Prototype* adalah pengembangan yang cepat dan pengujian terhadap model kerja (*prototipe*) dari aplikasi baru melalui proses interaksi dan berulang-ulang yang biasa digunakan ahli sistem informasi dan ahli bisnis. *Prototype* disebut juga desain aplikasi cepat (*rapid application design/RAD*) karena menyederhanakan dan mempercepat desain sistem.

#### ***H. Teknik Pengujian Sistem***

Pengujian sistem merupakan proses pengeksekusi sistem perangkat lunak untuk menentukan apakah sistem perangkat lunak tersebut cocok dengan spesifikasi sistem dan berjalan dengan lingkungan yang diinginkan. Pengujian sistem sering diasosiasikan dengan pencarian *bug*, ketidak sempurnaan program, kesalahan pada baris program yang menyebabkan kegagalan pada eksekusi sistem perangkat lunak.

Pada penelitian ini akan digunakan sistem *Black Box testing*, yaitu menguji perangkat lunak dari segi spesifikasi fungsional tanpa mengui desain dan kode program. Pengujian dimaksudkan untuk mengetahui apakah fungsi-fungsi, masukan

dan keluaran dari perangkat lunak sesuai dengan spesifikasi yang dibutuhkan (Rosa dan Shalahuddin, 2011).

## I. Rancangan Tabel Pengujian

Adapun rancangan tabel pengujian pada penelitian ini adalah sebagai berikut :

Tabel III.1 : kasus yang akan diuji

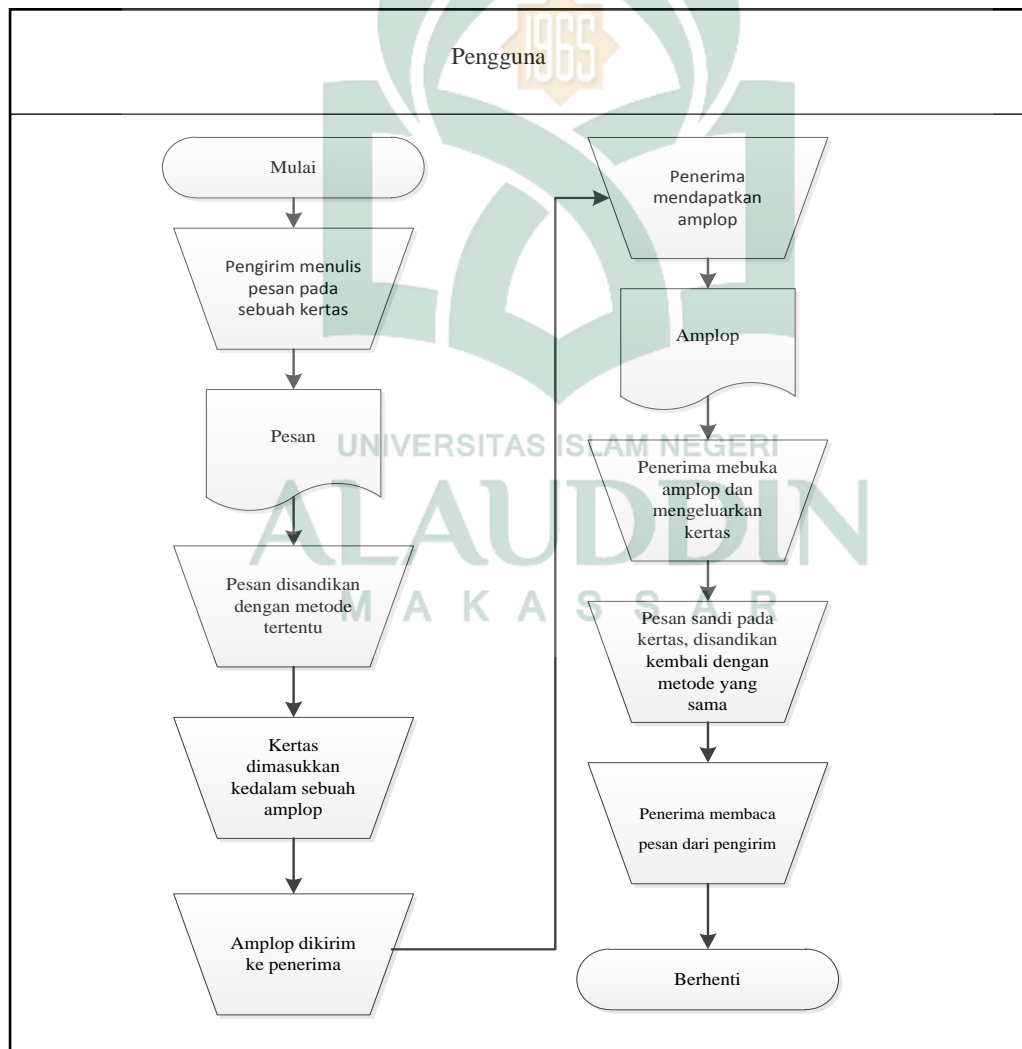
Kasus yang akan diuji
Apakah proses <i>load file audio</i> dan <i>image</i> berhasil?
Apakah proses <i>enkripsi</i> dan <i>dekripsi</i> berhasil?
Apakah proses <i>embedding</i> dan <i>extracting</i> berhasil?
Apakah proses <i>sharing file image</i> berhasil?
Media apa saja yang dapat digunakan untuk <i>sharing file</i> ?

## BAB IV

### ANALISIS DAN PERANCANGAN SISTEM

#### A. Analisis Sistem yang Sedang Berjalan

Proses pengacuan dan penyisipan sebuah pesan dapat dilihat pada *flowmap* diagram berikut :



Gambar IV.1 : *Flowmap* sistem yang berjalan

## ***B. Analisis Sistem yang diusulkan***

Analisis sistem adalah penguraian dari suatu sistem yang utuh ke dalam bagian-bagian komponennya dengan maksud untuk mengidentifikasi dan mengevaluasi permasalahan. Bagian analisis ini terdiri atas analisis masalah, analisis kebutuhan sistem, dan analisis kelemahan sistem.

### **1. Analisis Masalah**

Aplikasi steganografi merupakan aplikasi menyembunyi pesan kedalam sebuah media, dimana pesan akan disembunyikan ke dalam sebuah gambar. Terdapat dua fungsi utama dalam aplikasi ini, yaitu encoding dan decoding. Fungsi encoding berfungsi menyembunyikan/menanamkan pesan suara yang terenkripsi kedalam gambar berformat JPEG. Kemudian fungsi decoding, mengeluarkan pesan suara yang tersembunyi di dalam gambar, dan dikembalikan menjadi pesan suara yang utuh. Namun pada proses encoding, sebelum pesan disembunyikan, pesan tersebut dienkripsi terlebih dahulu, begitu pula dengan proses decoding, setelah pesan diekstrak, akan didekripsi menjadi pesan suara yang utuh.

### **2. Analisis Kebutuhan Sistem**

#### **a. Kebutuhan Antarmuka (*Interface*)**

- 1) Aplikasi akan menampilkan fungsi untuk menyembunyikan pesan suara ke dalam gambar dan fungsi untuk mengembalikan pesan suara yang tersembunyi pada gambar.

- 2) Aplikasi akan menampilkan panduan penggunaan aplikasi.
- 3) Aplikasi yang dibangun akan mempunyai antarmuka yang *familiar* dan mudah digunakan bagi pengguna.

b. Kebutuhan Data

- 1) Format gambar
- 2) Ukuran/*Resolusi* gambar
- 3) Format pesan suara
- 4) Kapasitas pesan suara

c. Kebutuhan Fungsional

Kebutuhan fungsional merupakan penjelasan proses fungsi yang berupa penjelasan secara terinci setiap fungsi yang digunakan untuk menyelesaikan masalah.

Fungsi-fungsi yang dimiliki oleh aplikasi ini adalah sebagai berikut:

- 1) Menampilkan fungsi encoding, yaitu proses menanamkan/meneyembunyikan pesan suara kedalam gambar.
- 2) Menampilkan fungsi decoding, yaitu proses mengekstrak/mengeluarkan gambar menjadi pesan suara yang utuh.
- 3) Mengirimkan gambar yang telah ditanamkan pesan suara.
- 4) Menampilkan panduan tentang cara penggunaan aplikasi.

3. Analisis Kelemahan

Aplikasi steganografi ini merupakan aplikasi yang berjalan pada *platform* android yang memberikan kemudahan dalam melakukan pengiriman



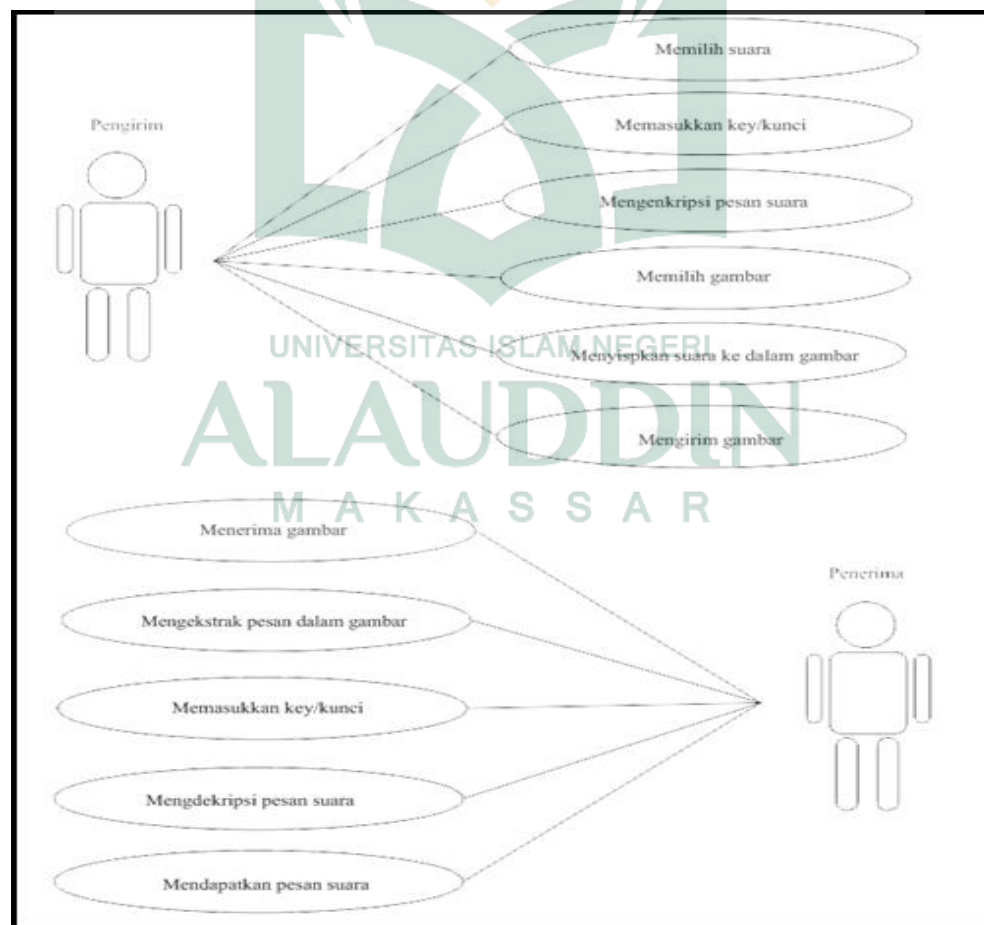
data yang bersifat rahasia dengan cara menyembunyikan ke dalam gambar.

Namun aplikasi ini hanya dapat menyembunyikan pesan yang berupa suara.

### C. *Percangan Sistem*

#### 1. *Use Case Diagram*

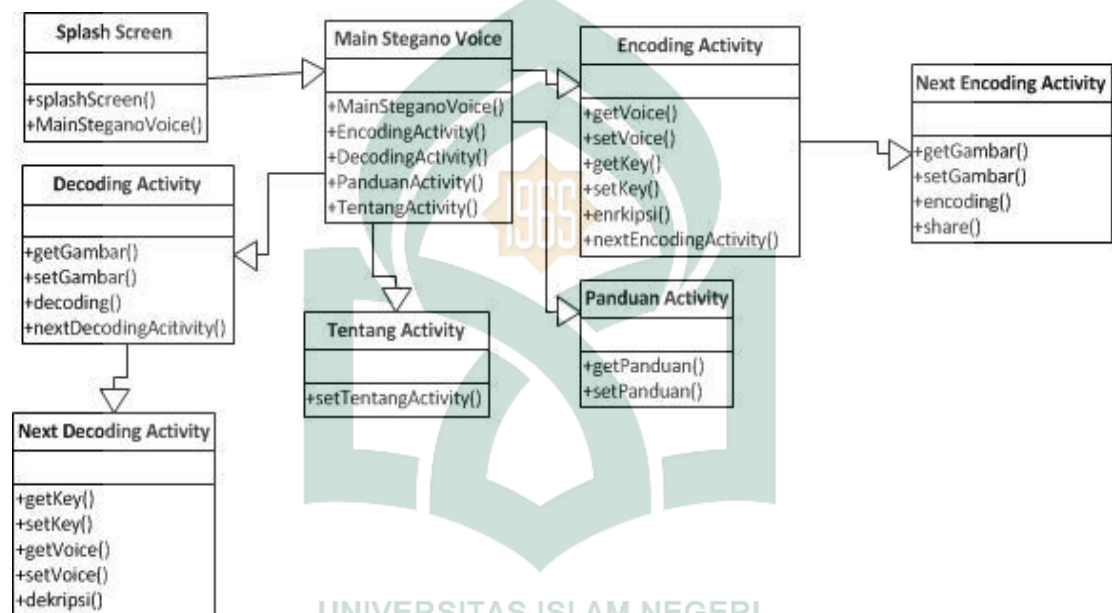
*Use case diagram* merupakan gambaran skenario dari interaksi antara pengguna dengan sistem. *Use case diagram* menggambarkan hubungan antara aktor dan kegiatan yang dapat dilakukannya terhadap aplikasi.



Gambar IV.2 : *use case diagram*

## 2. Class Diagram

*Class Diagram* merupakan diagram yang menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem.

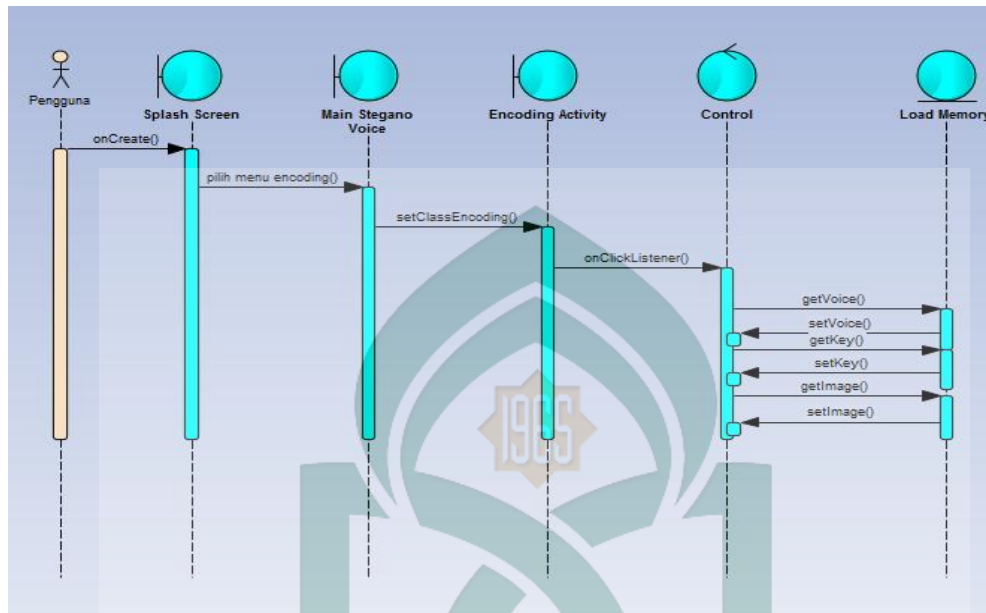


Gambar IV.3 : *Class Diagram*

## 3. Sequence Diagram

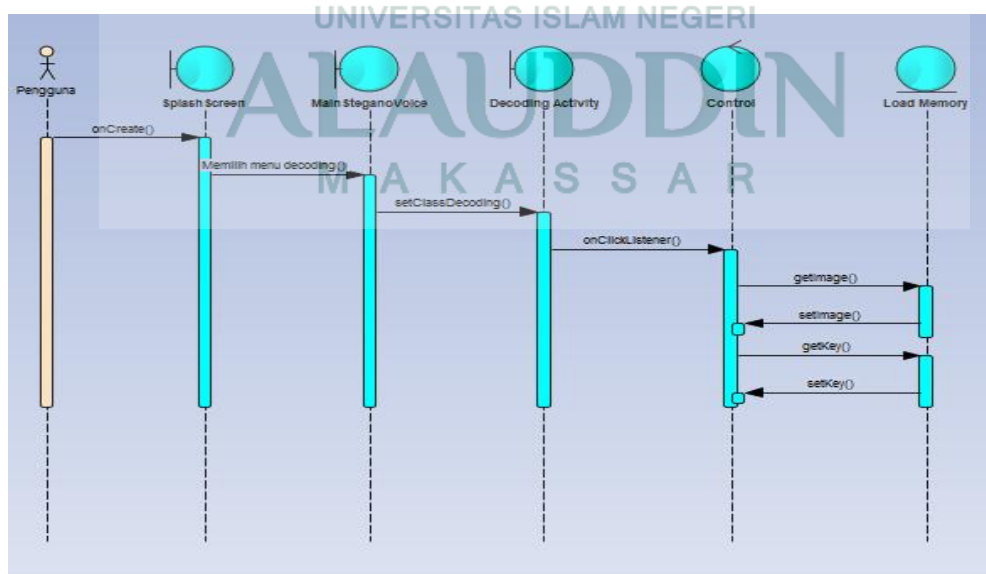
*Sequence diagram* menggambarkan interaksi antar objek di dalam dan di sekitar sistem berupa *message* terhadap waktu. Pembuatan *sequence diagram* bertujuan agar perancangan aplikasi lebih mudah dan terarah. Interaksi-interaksi yang terjadi dalam aplikasi steganografi adalah:

a. *Sequence Diagram Menu Encoding*



Gambar IV.4 : *sequence diagram menu encoding*

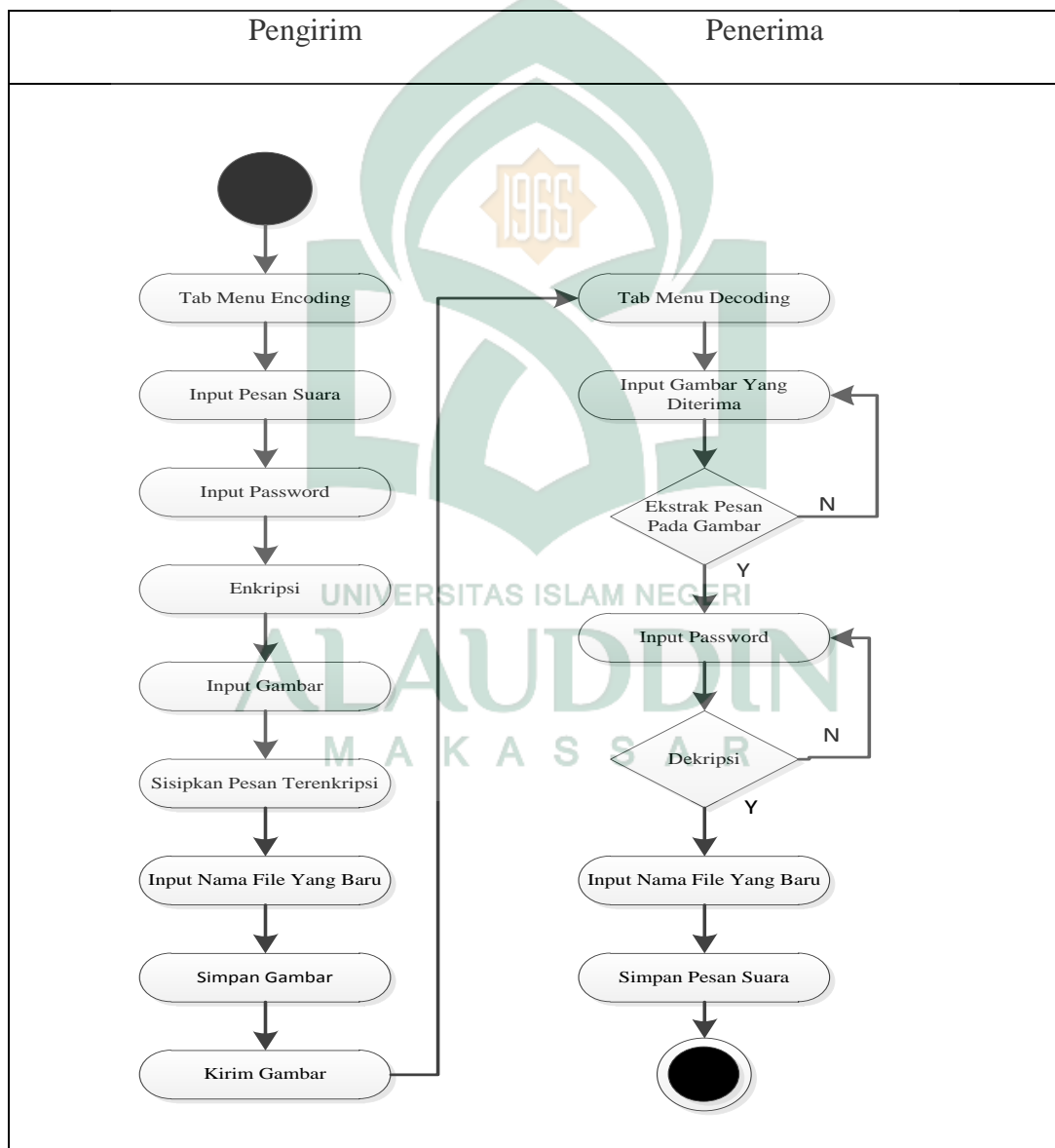
b. *Sequence Diagram Menu Decoding*



Gambar IV.5 : *sequence diagram menu decoding*

#### 4. Activity Diagram

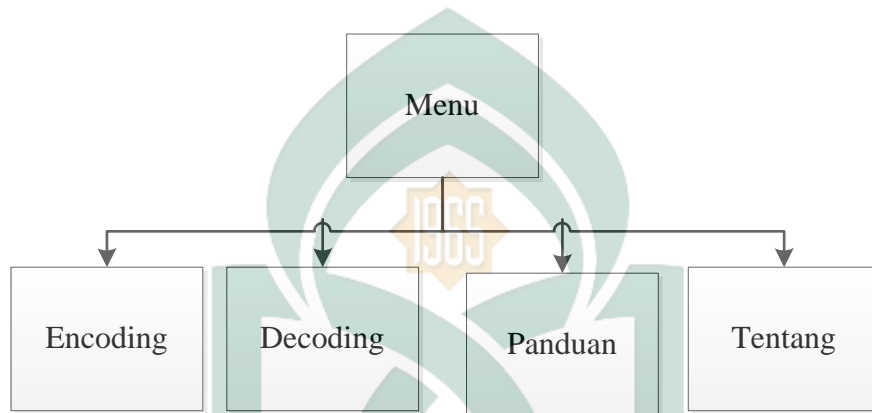
Activity diagram memodelkan workflow proses bisnis dan urutan aktivitas dalam sebuah proses. Activity diagram memudahkan untuk membantu memahami keseluruhan proses pada sistem.



Gambar IV.6 : Activity Diagram

## 5. Struktur Navigasi

Aplikasi Steganografi menggunakan struktur navigasi *Hierarchical Model*, di mana menu utama adalah pusat navigasi yang merupakan penghubung ke semua fitur pada aplikasi



Gambar IV.7 : Struktur Navigasi

Dari struktur navigasi ini, perpindahan antar fitur yang tersedia dapat dilakukan melalui menu.

## 6. Perancangan Antarmuka (*interface*)

Perancangan antarmuka merupakan aspek penting dalam perancangan aplikasi, karena berhubungan dengan tampilan dan interaksi yang memudahkan user dalam menggunakannya. Adapun rancangan antarmuka pada aplikasi ini sebagai berikut:

a. Perancangan Antarmuka *Splash Screen*



Gambar IV.8 : Desain antarmuka *splash screen*

Keterangan gambar :

1) Gambar

Akan dibuat gambar untuk splash screen aplikasi.

b. Perancangan Antarmuka Menu Utama



Gambar IV.9 : Desain antarmuka menu utama

Keterangan Gambar :

1) *Text*

Akan dibuat berisikan teks *Steganovoice*

2) Tabview dan gambar

Akan dibuat berisikan *icon encoding*

3) Tabview dan gambar

Akan dibuat berisikan *icon decoding*

4) Tabview dan gambar

Akan dibuat berisikan *icon panduan*

5) Tabview dan gambar

Akan dibuat berisikan *icon profil*

6) Konten

Akan dibuat berisikan konten *steganovoice*

c. Perancangan Antarmuka Menu *Encoding*

1		
2		
3		
4		
5		
6		
7		
8	9	

Gambar IV.10 : Desain antarmuka menu *encoding*

Keterangan gambar :

1. *Edit text*

Akan dibuat dengan berisikan *file* suara yang akan disembunyikan.

2. *Button*

Akan dibuat dengan berisikan tombol untuk meload *file* suara.

3. *Edit text*

Akan dibuat berupa inputan untuk kunci enkripsi.

4. *Button*

Akan dibuat dengan berisikan tombol untuk melakukan enkripsi.

5. *Image*

Akan dibuat dengan berisikan gambar yang akan dijadikan *cover*.

6. *Button*

Akan dibuat dengan berisikan tombol *embed* .

7. *Edit text*

Akan dibuat dengan berisikan nama file gambar yang telah disisipi pesan.

8. *Button*

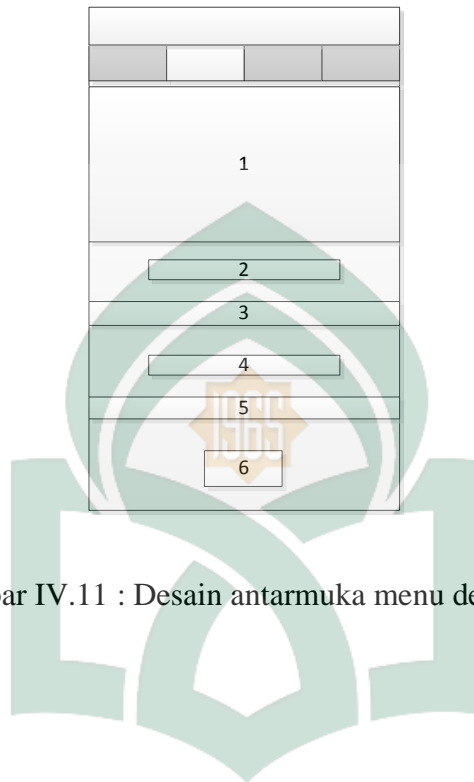
Akan dibuat dengan berisikan tombol *save*.

9. *Button*

Akan dibuat dengan berisikan tombol *share*.



d. Perancangan Antarmuka *Menu Decoding*



Gambar IV.11 : Desain antarmuka menu decoding

Keterangan gambar :

1. *Image*

Akan dibuat dengan berisikan gambar yang akan di *extract*.

2. *Button*

Akan dibuat dengan berisikan tombol *extract*.

3. *Edit Text*

Akan dibuat dengan berisikan kunci dekripsi.

4. *Button*

Akan dibuat dengan berisikan tombol dekripsi.

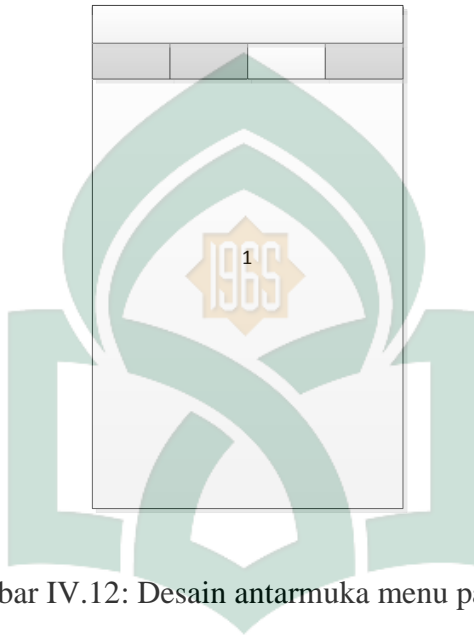
5. *Edit Text*

Akan dibuat dengan berisikan nama *file* suara yang akan disimpan.

## 6. *Button*

Akan dibuat dengan berisikan tombol *save*.

### e. Perancangan Antarmuka Menu Panduan



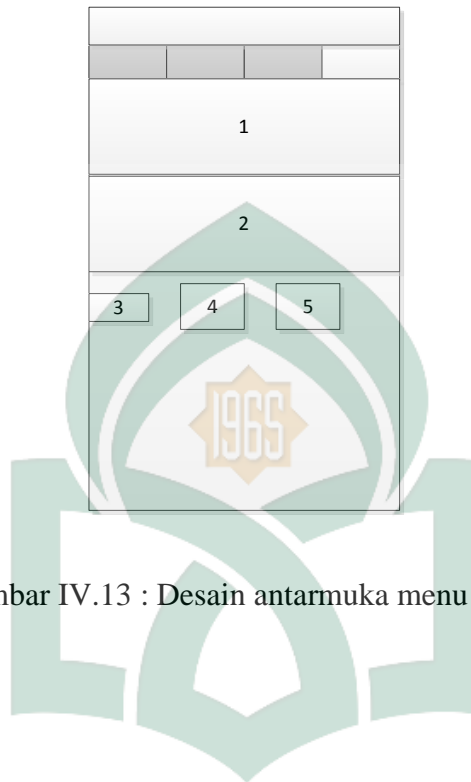
Gambar IV.12: Desain antarmuka menu panduan

Keterangan gambar : UNIVERSITAS ISLAM NEGERI

#### 1. Webview

Akan dibuat dengan berisikan teks panduan.

f. Perancangan Antarmuka Menu Profil



Gambar IV.13 : Desain antarmuka menu profil

Keterangan gambar :

1. Text

Akan dibuat dengan berisikan teks ucapan terima kasih

2. Image

Akan dibuat dengan berisikan gambar profil penulis

3. Text

Akan dibuat dengan berisikan teks kontak

4. Image

Akan dibuat dengan berisikan icon facebook

5. Image

Akan dibuat dengan berisikan *icon Google*.

## BAB V

### IMPLEMENTASI DAN ANALISIS HASIL

#### A. Implementasi

##### 1. Interface

###### a. Antarmuka *SplashScreen*

Antarmuka *splashscreen* menampilkan gambar *splashscreen* aplikasi selama 3 detik, dan setelah 3 detik akan menampilkan antarmuka *Encoding*.



Gambar V.1 : Antarmuka *splashscreen*.

###### b. Antarmuka *Encoding*

Antarmuka *encoding* akan tampil setelah *splashscreen*. Antarmuka *encoding* terdiri dari tombol *load file* suara, *edit text* untuk nama *file* suara, *edit text* untuk memasukkan *password*, tombol enkripsi untuk proses enkripsi *file* suara, *image view* untuk meload *file* gambar, tombol sisipkan untuk menyisipkan *file* suara terenkripsi

ke dalam gambar, *edit text* nama file gambar untuk memasukkan nama *file* gambar, tombol simpan untuk menyimpan *file* gambar yang telah disisipi pesan suara terenkripsi, dan tombol bagikan untuk mengirim *file* gambar.



Gambar V.2 : Antarmuka menu *encoding*.

### c. Antarmuka *Decoding*

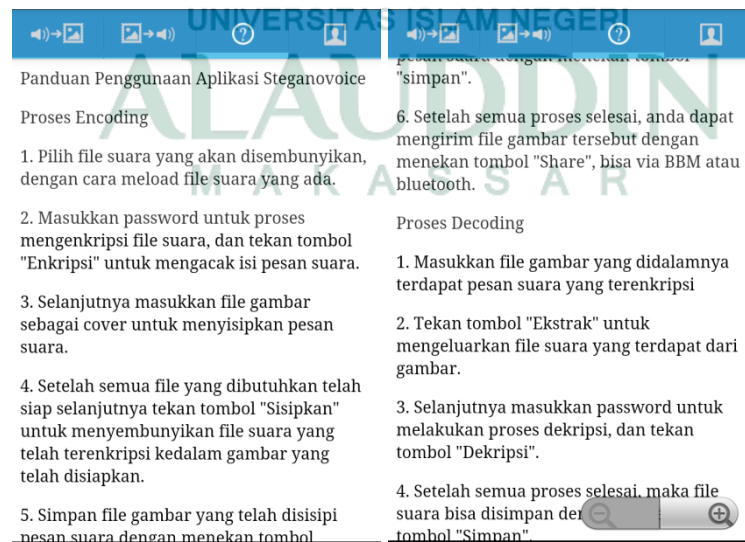
Antarmuka *decoding* terdiri dari *image view* untuk meload *file* gambar, tombol ekstrak untuk mengekstrak pesan suara terenkripsi dari *file* gambar, *edit text password* untuk memasukkan *password* dekripsi, tombol dekripsi untuk mendekripsi *file* suara, *edit text* nama *file* suara untuk memasukkan nama *file* suara dan tombol simpan untuk menyimpan *file* suara.



Gambar V.3 : Antarmuka menu *decoding*.

#### d. Antarmuka Panduan

Antarmuka panduan menampilkan *web view* yang berisi panduan penggunaan aplikasi *Steganovoice*.



Gambar V.4 : Antarmuka menu panduan.

e. Antarmuka Profil

Antarmuka profil menampilkan profil ucapan terima kasih penulis kepada semua yang menginstal aplikasi ini, profil penulis, serta kontak penulis.



Gambar V.5 : Antarmuka menu profil.

## B. Pengujian

Pengujian sistem merupakan proses pengeksekusian sistem perangkat lunak untuk menentukan apakah sistem tersebut cocok dengan spesifikasi sistem dan berjalan di lingkungan yang diinginkan. Pengujian sistem sering diasosiasikan dengan pencarian bug, ketidaksempurnaan program, kesalahan pada program yang menyebabkan kegagalan pada eksekusi sistem perangkat lunak.

Pengujian dilakukan dengan menguji setiap proses dan kemungkinan kesalahan yang terjadi untuk setiap proses. Pada penelitian ini menggunakan metode pengujian *black box*, yaitu menguji perangkat lunak dari segi spesifikasi fungsional tanpa mengui desain dan kode program. Pengujian dimaksudkan untuk mengetahui apakah fungsi-fungsi, masukan dan keluaran dari perangkat lunak sesuai dengan spesifikasi yang dibutuhkan.

#### 1. Prosedur Pengujian

Persiapan yang dilakukan pada pengujian ini adalah sebagai berikut:

- a. Menyiapkan *smartphone* dengan sistem operasi Android.
- b. Menginstal Aplikasi Penyembunyi Pesan Suara Terenkripsi ke *smartphone* Android.
- c. Melakukan proses pengujian.
- d. Mencatat hasil pengujian.

#### 2. Hasil Pengujian

- a. Pengujian Menu *Encoding*

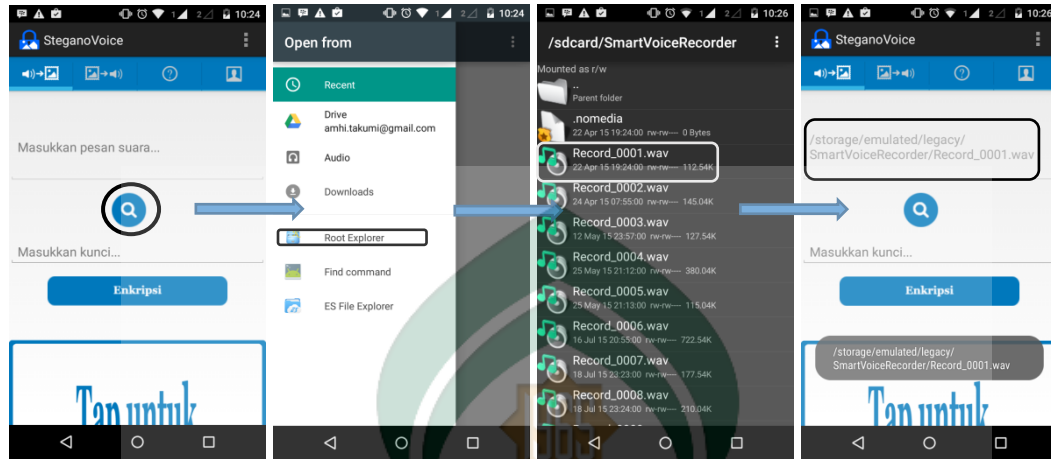
Tabel V.1 : Tabel pengujian menu *encoding*

Kasus dan hasil uji			
Data masukan	Yang diharapkan	Pengamatan	Kesimpulan
Menekan <i>button load</i> suara	Akan tampil pilihan media penyimpanan suara dan menampilkan nama dan lokasi <i>file</i> pada	Pilihan aksi sesuai yang diharapkan	Berhasil



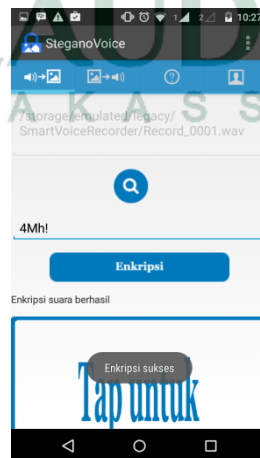
	<i>edit text</i>		
Menekan <i>edit text password</i>	Menginput <i>password</i> untuk proses enkripsi.	Pilihan aksi sesuai yang diharapkan	Berhasil
Menekan tombol Enkripsi	Melakukan proses enkripsi pesan suara dan memunculkan notifikasi berupa <i>toast</i> .	Pilihan aksi sesuai yang diharapkan	Berhasil
Menekan kotak <i>image view</i>	Akan tampil pilihan media penyimpanan gambar dan akan menampilkan gambar pada kotak <i>image view</i>	Pilihan aksi sesuai yang diharapkan	Berhasil
Menekan button sisipkan	Melakukan proses penyisipan/penanaman pesan suara terenkripsi	Pilihan aksi sesuai yang diharapkan	Berhasil
Menekan <i>edit text nama file</i>	Menginput nama <i>file</i> gambar yang baru	Pilihan aksi sesuai yang diharapkan	Berhasil
Menekan button simpan	Akan meyimpan gambar yang baru kedalam folder yang telah disiapkan	Pilihan aksi sesuai yang diharapkan	Berhasil
Menekan button bagikan	Melakukan pengiriman file dengan beberapa pilihan media	Pilihan aksi sesuai yang diharapkan	Berhasil

Adapun hasil dari pengujian diatas dapat dilihat pada gambar *screenshoot* berikut :



Gambar V.6 : Hasil pengujian *button load* suara

Pada gambar V.6 di atas, menampilkan fungsi *button load* suara yang jika dipilih akan muncul beberapa media penyimpanan *audio*, dan setelah memilih file *audionya*, maka pada *edit text* nama pesan suara akan muncul lokasi dan nama file *audio* yang terpilih.



Gambar V.7 : Hasil pengujian *button enkripsi*

Pada gambar V.7, menampilkan fungsi *button* enkripsi dimana terlebih dahulu edit text password diisi dengan password enkripsi, kemudian menekan button enkripsi yang berfungsi untuk mengenkripsi pesan suara. Pesan suara yang berhasil terenkripsi akan menampilkan notifikasi berupa toast.



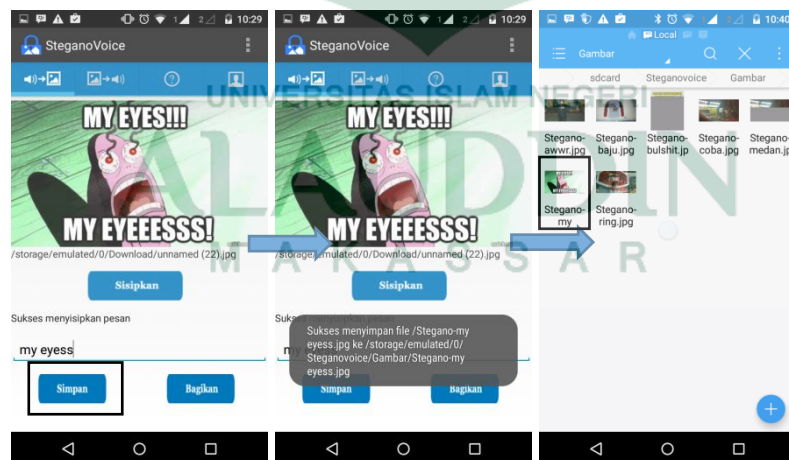
Gambar V.8 : Hasil pengujian *image view*

Pada gambar V.8 di atas, menampilkan fungsi meload gambar dengan cara memilih *image view* “Tap untuk memilih gambar” maka akan menampilkan beberapa media penyimpanan gambar. Kemudian gambar yang terpilih akan muncul pada kotak *image view*, dan akan muncul notifikasi lokasi *file* gambar berada dan nama *file* gambar dalam bentuk *toast*.



Gambar V.9 : Hasil pengujian *button* sisipkan

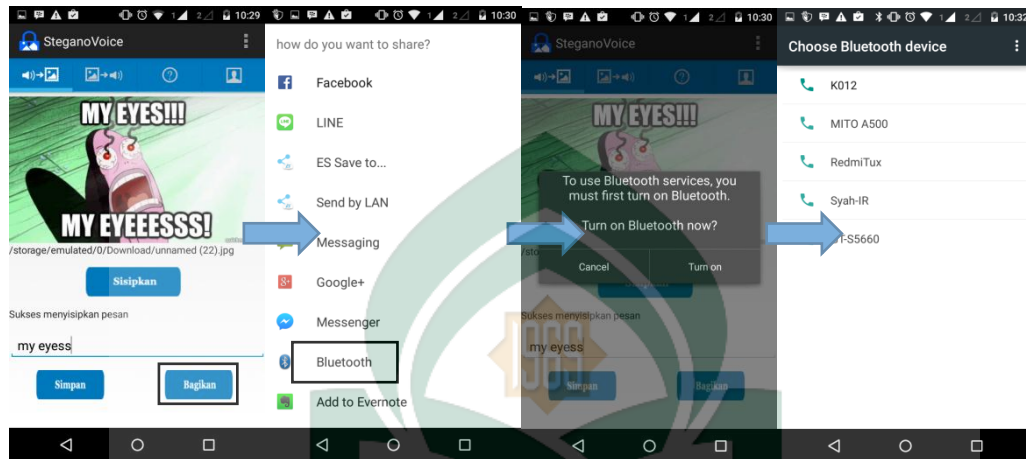
Pada gambar V.9 di atas, menampilkan fungsi *button* sisipkan yang berfungsi untuk menyisipkan/menanamkan pesan suara terenkripsi ke dalam gambar. Notifikasi berupa toast akan muncul ketika proses penyisipan selesai.



Gambar V.10 : Hasil pengujian *button* simpan

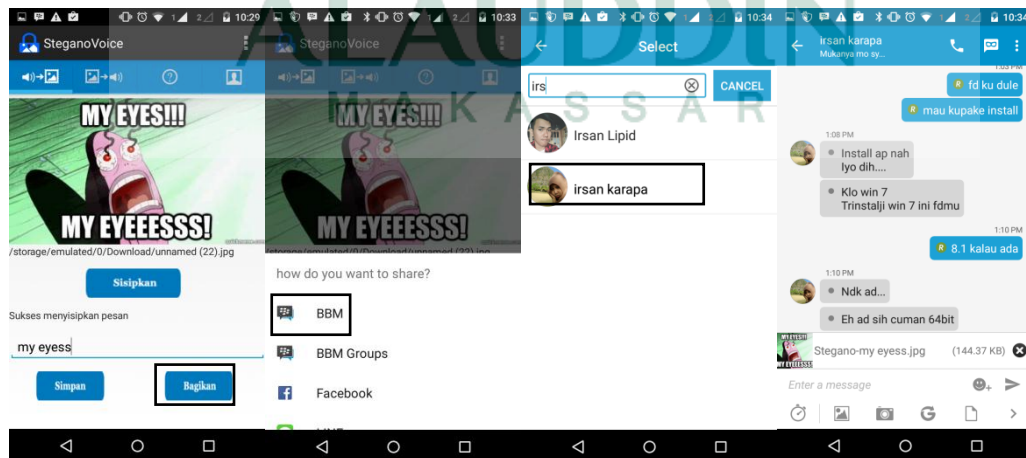
Pada gambar V.10 di atas, menampilkan fungsi *button* simpan dimana terlebih dahulu mengisi *edit text* yang telah disiapkan untuk nama *file* gambar yang baru,

setelah itu memilih *button* simpan untuk menyimpan *file* gambarnya ke dalam folder *Steganovoice/gambar*, dan notifikasi berupa *toast* akan muncul.



Gambar V.11 : Hasil pengujian *button* bagikan via *bluetooth*

Pada gambar V.11 di atas, menampilkan fungsi *button* bagikan. Saat *button* bagikan dipilih, maka akan muncul beberapa media pengiriman *file*, pada gambar di atas menggunakan *bluetooth* untuk proses pengirimannya.



Gambar V.12 : Hasil pengujian *button* bagikan via *BBM*

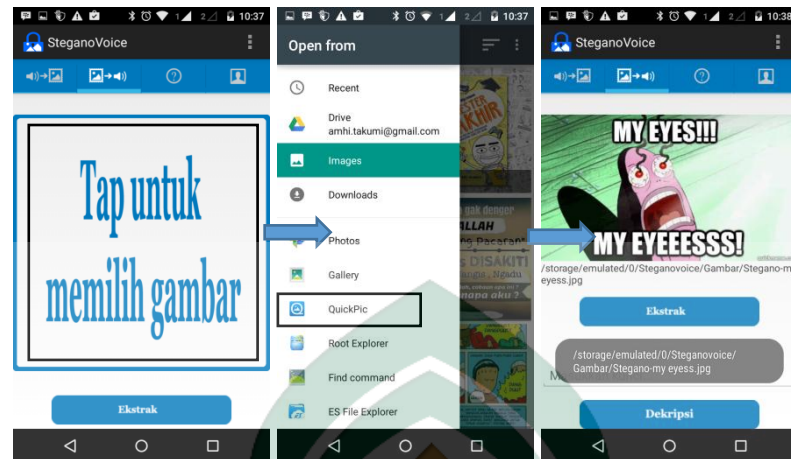
Pada gambar V.12 di atas, menampilkan fungsi *button* bagikan. Saat *button* bagikan dipilih, maka akan muncul beberapa media pengiriman *file*, pada gambar di atas menggunakan BBM untuk proses pengirimannya.

b. Pengujian menu *decoding*

Tabel V.2 : Tabel menu *decoding*

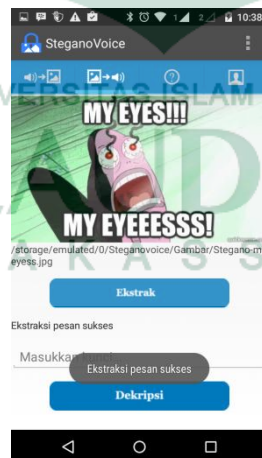
Kasus dan hasil uji			
Data masukan	Yang diharapkan	Pengamatan	Kesimpulan
Menekan kotak <i>image view</i>	Akan tampil pilihan media penyimpanan gambar dan menampilkan gambar terpilih pada <i>image view</i> .	Pilihan aksi sesuai yang diharapkan	Berhasil
Menekan button ekstrak	Melakukan proses ekstrak pesan suara terenkripsi dari gambar.	Pilihan aksi sesuai yang diharapkan	Berhasil
Menekan <i>edit text password</i>	Menginput password untuk proses dekripsi.	Pilihan aksi sesuai yang diharapkan	Berhasil
Menekan tombol dekripsi	Melakukan proses dekripsi pesan suara.	Pilihan aksi sesuai yang diharapkan	Berhasil
Menekan <i>edit text nama file</i>	Menginput nama <i>file</i> suara yang baru	Pilihan aksi sesuai yang diharapkan	Berhasil
Menekan button simpan	Akan meyimpan suara yang baru kedalam folder yang telah disiapkan	Pilihan aksi sesuai yang diharapkan	Berhasil

Adapun hasil dari pengujian diatas dapat dilihat pada gambar *screenshot* berikut :



Gambar V.13 : Hasil pengujian *image view*

Pada gambar V.13 di atas, menampilkan fungsi *image view*, dimana saat dipilih akan menampilkan beberapa media penyimpanan gambar, dan setelah memilih gambar, maka gambar yang terpilih akan muncul pada *image view*.



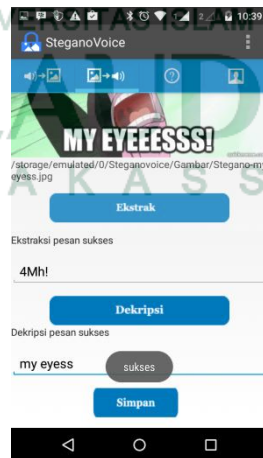
Gambar V.14 : Hasil pengujian *button ekstrak*

Pada gambar V.14 di atas, menampilkan fungsi button ekstrak yang berfungsi untuk mengekstrak pesan yang tersembunyi di dalam gambar. Notifikasi berupa *toast* akan muncul jika proses ekstraksi selesai.



Gambar V.15 : Hasil pengujian *button* dekripsi

Pada gambar V.15 di atas, menampilkan fungsi *button* dekripsi. Pada *edit text password*, terlebih dahulu memasukkan *password* untuk proses dekripsinya. Kemudian pilih *button* dekripsi untuk mendekripsi isi pesannya. Notifikasi berupa *toast* akan muncul jika proses dekripsi selesai.



Gambar V.16 : Hasil pengujian *button* simpan

Pada gambar V.16 di atas, menampilkan fungsi *button* simpan dimana terlebih dahulu mengisi *edit text* yang telah disiapkan untuk nama *file* gambar yang baru,

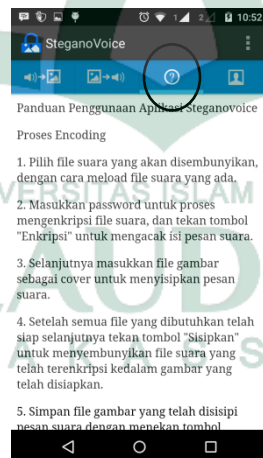


setelah itu memilih *button* simpan untuk menyimpan *file* gambarnya ke dalam folder *Steganovoice*/suara, dan notifikasi berupa *toast* akan muncul.

### c. Pengujian Menu Panduan

Tabel V.3 : Tabel menu panduan

Kasus dan hasil uji			
Data masukan	Yang diharapkan	Pengamatan	Kesimpulan
Menekan <i>tab</i> panduan	Akan tampil menu panduan yang berisi panduan penggunaan aplikasi.	Pilihan aksi sesuai yang diharapkan	Berhasil



Gambar V.17 : Hasil pengujian *tab* panduan

Pada gambar V.17 di atas, menampilkan halaman panduan saat *tab* panduan dipilih.

## d. Pengujian Menu Profil

Tabel V.4 : Tabel menu panduan

Kasus dan hasil uji			
Data masukan	Yang diharapkan	Pengamatan	Kesimpulan
Menekan <i>tab</i> profil	Akan tampil menu profil yang berisi konten profil.	Pilihan aksi sesuai yang diharapkan	Berhasil

Gambar V.18 : Hasil pengujian *tab* profil

Pada gambar V.18 di atas, menampilkan halaman profil ketika *tab* profil dipilih.

e. Tabel hasil *encoding*

Tabel V.4 : Tabel hasil encoding

No	Gambar		Keterangan
	Sebelum <i>Encoding</i>	Setelah <i>Encoding</i>	
1			Gambar tetap tapi kapasitas berubah
Kapasitas	1,413,628 bytes	1,562,161 bytes	
2			Gambar tetap tapi kapasitas berubah
Kapasitas	32,578 bytes	147,831 bytes	
3			Gambar tetap tapi kapasitas berubah
Kapasitas	1,139,277 bytes	1,254,530 bytes	
4			Gambar tetap tapi kapasitas berubah
Kapasitas	1,115,507 bytes	1,230,760 bytes	

## **BAB VI**

### **PENUTUP**

#### ***A. Kesimpulan***

Dari pembahasan yang telah dijelaskan pada bab-bab sebelumnya, dapat ditarik kesimpulan bahwa aplikasi ini berhasil melakukan proses enkripsi dan dekripsi pesan suara dengan menggunakan algoritma *serpent*. Aplikasi ini juga berhasil melakukan proses penyisipan dan ekstraksi pesan suara dengan mengimplementasikan teknik steganografi metode EOF (*End of File*) yang dapat berjalan pada sistem operasi Android. Berdasarkan hasil pengujian, gambar hasil enkripsi tidak mengalami perubahan yang signifikan tetapi mengalami perubahan pada kapasitas yang bertambah pada saat setelah disisipkan pesan. Namun, prosesnya berjalan lambat, terutama saat melakukan ekstraksi pesan, dikarenakan proses pencarian datanya dilakukan perbit dalam bit-bit gambar. Aplikasi ini berhasil melakukan proses pengiriman hanya dengan menggunakan *Bluetooth* dan BBM.

#### ***B. Saran***

Aplikasi Steganografi Penyembunyi Pesan Suara Terenkripsi ini masih jauh dari kesempurnaan, untuk menciptakan sebuah aplikasi yang baik tentu perlu dilakukan pengembangan baik dari sisi manfaat maupun dari sisi kerja sistem, berikut beberapa saran bagi yang ingin mengembangkan aplikasi yang mungkin dapat menambah nilai dari aplikasi nantinya:

1. Gunakan media lain dalam penyembunyian file suara, media gambar kurang tepat digunakan karena perbedaan ukuran kapasitas yang cukup besar.
2. Penggunaan metode lain pada proses steganografinya selain metode *end of file*.
3. Penggunaan algoritma lain pada proses kriptografinya selain algoritma *serpent*
4. Pada penggunaan metode steganografi LSB (dalam penelitian ini), carikan solusi agar penggunaan metode LSB dapat diterapkan, karena terdapat perbedaan kapasitas yang cukup besar antara suara dan gambar.

## Daftar Pustaka

- Ayuningsih, Iin Dwi. “Aplikasi Penyembunyi Pesan Pada *File* Citra Dengan Menggunakan Metode Steganografi *EOF (End Of File)* Berbasis *Mobile*”. *Skripsi*. Makassar : Universitas Islam Negeri Makassar, 2014.
- Dony. “Arsitektur Sistem Operasi Android”, *Blog Bondhezh Zone*. <http://bondhezh.wordpress.com/2011/11/15/arsitektur-sistem-operasi-android/> (2011, November 15).
- Huda, Arif Akbarul, 24 *JAM!! Pintar Pemrograman Android*. <http://www.omyib.com> (21 Agustus 2014).
- Irawan, Joseph Dedy dan Emmalia Adriantantri. “Steganografi Untuk Menyembunyikan Suara Dengan Smart Card Sebagai Kunci Enkripsi”. *Skripsi*. Malang: Institut Teknologi Nasional Malang, 2010.
- Lepank. “pengertian aplikasi menurut beberapa ahli”. *Blog Lepank*. <http://www.lepank.com/2012/08/pengertian-aplikasi-menurut-beberapa.html> (20 Agustus 2014)
- Narbuko, Drs. Cholid dan Drs. H. Abu Achmadi. *Metodologi Penelitian*. Jakarta : PT. Bumi Aksara
- Nugroho, Bayu Kristian. “*Aplikasi Enkripsi Sms Pada Telepon Selular Berbasis J2Me Dengan Metode Vigenere Cipher*”. *Skripsi*. Semarang: Universitas Diponegoro Semarang, 2010.
- Pemerintah Kota Medan. “Undang-Undang Republik Indonesia Nomor 36 Tahun 1999 Pasal 41”. Official Website Pemerintah Kota Medan. [http://www.pemkomedan.go.id/uuti/uu\\_361999.php](http://www.pemkomedan.go.id/uuti/uu_361999.php) (24 Agustus 2014).
- “Penelitian Kualitatif”. *Wikipedia the Free Encyclopedia*. [http://id.wikipedia.org/wiki/Penelitian\\_kualitatif](http://id.wikipedia.org/wiki/Penelitian_kualitatif) (3 November 2013)
- Perdana, Jaka. “41 Juta Masyarakat Indonesia Miliki *Smartphone* 95% nya digunakan di Rumah”. *Blog Marketeers*. <http://www.the-marketeers.com/archives/41-juta-masyarakat-indonesia-miliki-smartphone-95nya-digunakan-di-rumah.html> (2013, Oktober 29).

- Pressman, Roger. S, *Software Engineering: A Practitioner's Approach, Fifth Edition. United State: McGraw Hill Companies, Inc*, 2001.
- Putri, Anggi Alisia. "*Studi dan Implementasi Enkripsi Pengiriman Pesan Suara dengan Algoritma Serpent*". Skripsi. Bandung: Institut Teknologi Bandung, 2009.
- Rosa dan M. Shalahuddin. Modul Pembelajaran Rekayasa Perangkat Lunak (Terstruktur dan Berorientasi Objek). Bandung: Modula, 2011.
- Rosenberg, D. *Use Case Driven Object Modeling With UML*. New York: Apress, 2007.
- Sadikin, Rifki. *Kriptografi Untuk Keamanan Jaringan*. Yogyakarta: C.V ANDI OFFSET, 2012.
- "Sistem Operasi Android". *Wikipedia the Free Encyclopedia*. [http://id.wikipedia.org/wiki/Android\\_\(sistem\\_operasi\)](http://id.wikipedia.org/wiki/Android_(sistem_operasi)) (20 Juli 2014)
- Tiro, Muhammad Arif. *Penelitian : Skripsi, Tesis, dan Disertasi*. Makassar: Andira Publisher, 2011.
- Universitaspendidikan. "Pengertian White Box dan Contoh White Box Testing". *Blog Universitaspendidikan*. <http://universitaspendidikan.com/pengertian-white-box-dan-contoh-white-box-testing/> (21 Juni 2013)
- "Versi Android". *Wikipedia the Free Encyclopedia*. [http://id.wikipedia.org/wiki/Daftar\\_versi\\_Android](http://id.wikipedia.org/wiki/Daftar_versi_Android) (15 Agustus 2014)
- "Steganografi". *Wikipedia the Free Encyclopedia*. <http://id.wikipedia.org/wiki/Steganografi> (19 Januari 2014).